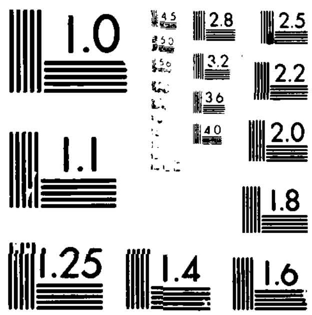
AD-A081 770 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A USER'S MANUAL FOR LINEAR CONTROL PROGRAMS ON IBM/360.(U)
DEC 79 B DESJARDINS

UNCLASSIFIED NL

1 OF 3
AD
A081770



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A USER'S MANUAL FOR LINEAR CONTROL
PROGRAMS ON IBM/360

by

Berthier Desjardins

December 1979

Thesis Advisor: D. E. Kirk

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 A User's Manual for Linear Control Programs on IBM/360.		5. TYPE OF REPORT & PERIOD COVERED 9 Master's Thesis, ~~December 1979~~
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Berthier Desjardins		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940 12 261		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940 11		[redacted] Dec[redacted]79
		[redacted] 260
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear Control Programs
IBM/360

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A linear control subroutine library was created and stored in a load module on Disk 02 of the IBM/360 of the Naval Postgraduate School.

This library consists of three groups of programs: transfer function subprograms; matrix manipulation and time response subprograms; and modern control design routines. The transfer function subprograms provide numerical aids for classical control

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

UNCLASSIFIED 251450
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. ABSTRACT Continued)

design techniques including root locus and frequency design methods. The matrix manipulation and time response routines allow the user to determine eigenvalues, find state transition matrices, evaluate resolvent matrices, perform several other matrix operations and determine and plot graphical time responses. The modern control design programs aid in solving Linear Quadratic Gaussian (LQG) problems and also provide the capability to investigate sensitivity and to de-couple multi-input multi-output systems.

This thesis is a user's manual for the library of control design programs. Applications, extensive documentation and numerous worked examples are included.

Approved for public release; distribution unlimited.

A User's Manual for Linear Control
Programs on IBM/360

by

Berthier Desjardins
Captain, Canadian Armed Forces
B.S. Math and Physics, College militaire royal de St. Jean, 1973
B.S.E.E., Naval Postgraduate School, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1979

Author Berthier Desjardins

Approved by: D E Kirk
Thesis Advisor

Alex Gerba Jr.
Second Reader

D E Kirk
Chairman, Department of Electrical Engineering

William M. Tolles
Dean of Science and Engineering

ABSTRACT

A linear control subroutine library was created and stored in a load module on Disk 02 of the IBM/360 of the Naval Postgraduate School.

This library consists of three groups of programs: transfer function subprograms; matrix manipulation and time response subprograms; and modern control design routines. The transfer function subprograms provide numerical aids for classical control design techniques including root locus and frequency design methods. The matrix manipulation and time response routines allow the user to determine eigenvalues, find state transition matrices, evaluate resolvent matrices, perform several other matrix operations and determine and plot graphical time responses. The modern control design programs aid in solving Linear Quadratic Gaussian (LQG) problems and also provide the capability to investigate sensitivity and to de-couple multi-input multi-output systems.

This thesis is a user's manual for the library of control design programs. Applications, extensive documentation and numerous worked examples are included.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A large number of computer programs are available to help today's control system engineers analyse and design increasingly complex systems. Most of these programs, however, are available in the form of listings only, so any one wishing to use them must punch the cards, compile and test the routines, modify them and, most of the time, load them everytime a problem is to be solved. Obviously this is not a very practical and efficient way of using computers.

The intent of this thesis was to facilitate the use of several of these programs by making them easily accessible to all users as a pre-compiled load module library. The features of the library were to be as follows:

- easy access to the subprograms
- only rudimentary knowledge of FORTRAN coding and card set up procedures required to use the subprograms
- good documentation readily available to the users (complete with subprogram descriptions, card set up procedures and worked out examples)
- an expandable and improvable library
- good priority categories (class A or B only) for quick turnaround.

Using these features as guidelines, a linear control subroutines library (LINCON) was created, tested and is now available to any user on the Naval Postgraduate School IBM/360.

The following chapters describe the computer procedures and present these linear control subprograms contained in the library. All the different aspects of deck preparation and job control cards are discussed. The most common error conditions that may occur while using the subprograms and the remedial actions to be taken are pointed out. The linear control subprograms are presented in a user-oriented fashion. They are first introduced by defining their purposes and indicating the general rules that apply. Then the subprograms are individually described. The input requirements and the output to be expected are presented in great detail. Several examples are worked out, complete with the control cards, the input data, the computer output and the interpretations of the results.

Note that the programming aspects of the work are not included in the presentation. Reference 1, the provenance of most of the subprograms that constitute the LINCON library, must be consulted in that regard, along with the actual listings of the subroutines. Also note that reference 1 can be used as an alternative source of information in using the subprograms.

However, Appendix A explains how the LINCON data sets were created and gives the job control cards required to modify, verify or erase the data sets. Information on how to recreate the library, should it become necessary, are given as well. Finally, Appendix B specifies the references from which the worked examples were taken.

II. COMPUTER PROCEDURE

The programs and subprograms described in [1], or modified versions, together with a few locally written programs were assembled in a load module to form a subroutine library. A private user disk space was allocated on Disk02 of the Naval Postgraduate School IBM/360 computer to hold the partitioned data set and library procedures were defined and cataloged so the library could be accessed by any computer user under OS Batch. The details on how to access the library and use the subprograms and subroutines are presented in the following paragraphs and a complete description of the data set, along with pertinent computer procedure information, is given in Appendix A.

The system was devised in such a way as to minimize the need for programming and provide the user with a convenient, flexible, easy-to-utilize tool for analysis and design of linear control systems. The programs and subprograms were kept as separate subroutines so one, or more, programs could be executed as a single job. The following gives a detailed description of the different methods of accessing the library as well as the cards necessary to run the programs under batch processing. For convenience, the major steps of the procedure are also reproduced in Section III as part of the subprograms presentation.

A. MODES OF OPERATION AND CONTROL CARDS

There are many different operating modes a user can employ to access a given computer library. Of these methods, three were determined to be most appropriate and are presented hereafter. It is pointed out that these procedures only apply to this specific set of programs, which was assigned the name LINCON (for linear control). Also observe that each mode implies the use of slightly different control card deck set ups. These differences are essential to the proper operation of the system under the selected mode of operation. Each line must be meticulously reproduced on the computer card and the order of appearance of the cards scrupulously respected.

1. Mode One

This mode applies when a user wants to execute only one of the subprograms for either single or multiple runs. Except for the subprograms named GTRESP, KALMAN and PRTLOC, which require exterior subroutines, all subprograms can be accessed using this method. Mode Two establishes the procedures that deal with the three special cases enumerated above.

For Mode One, the control cards must be:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD *
^^INCLUDE^SYSLIB(member)
/*
//GO.SYSIN^DD^*
```

data deck as described in Section III
for the subprogram "member"

```
/*
```

where "member" is the simple name defining a subprogram to be executed. For example, "include syslib(SERCOM)" would have to be typed on the appropriate card to access the subroutine library program called SERCOM.

2. Mode Two

The three special cases previously mentioned are accessed using this mode of operation. A different library procedure was created since GTRESP, KALMAN or PRTLOC might very well require special functions or inputs that vary as given parameters change. This situation does not significantly complicate the procedure and greatly adds to the system capability. Further justification and explanation are given in Section III along with the subprogram descriptions. Again under this second mode, the programs are to be accessed one at a time, either for single or multiple runs. The computer card deck set up to be provided is:

```
// (standard OS JOB card)
//^EXEC^LINCONF
//FORT.SYSIN^DD^*
        FORTRAN deck of user supplied subroutine as
        specified for GTRESP, KALMAN or PRTLOC
/*
//LINK.SYSIN^DD^*
```

```
^^INCLUDE^SYSLIB(member)
^^ENTRY^member
/*
//GO.SYSIN^DD^*
       data deck for "member" as described
       in Section III.
/*
```

where member is the actual name of the subprogram to be executed. In this case, it is either GTRESP, KALMAN or PRTLOC.

For example, "include syslib(KALMAN)" on the appropriate card, followed by "entry KALMAN" on the next card would cause the subprogram called KALMAN to be run.

3. Mode Three

This mode of operation permits the user to call more than one subprogram while executing a single job. Since this third option calls all the subprograms at the same time, a large amount of computer memory is required. The user must be aware that this increases the turnaround time. Nevertheless the method can still be very useful. For instance, a user who is not in a hurry could utilize this set up to obtain the solution to several simple problems which do not require modification of some parameters.

At this point the user is reminded that great care must be taken to correctly prepare the control and data decks. With an increased turnaround time, errors become costly and very frustrating.

When it is decided to use Mode Three, the following computer cards must be generated:

```
// (standard OS JOB card), TIME=5
//^EXEC^LINCON,REGION.GO=350K
^^ INCLUDE^SYSLIB(MAIN)
/*
//GO.SYSIN^DD^*
MEMBER 1
        data deck for member 1 as
        described in Section III
$
MEMBER 2
        data deck for member 2 as
        described in Section III
$
/*
```

where MEMBER 1, MEMBER 2, etc., are the defining names of the subprograms to be executed and start in column one. Note that again, as explained in Section III, the data deck pertaining to the same subprogram can be arranged either for single or multiple runs. The dollar sign, $, is a stop sign to be printed in column one. This dollar sign card must appear after the last data deck of each "member" to be executed under Mode Three.

B. ERROR CONDITIONS

When running programs, it is rather disappointing if results do not come out as expected. This in itself is a good

reason to always verify one last time that the control cards were punched correctly and the data deck was set up exactly as specified. Nonetheless, both neophytes and veterans do make mistakes and the purpose of this section is to outline some of the most common errors and show how to identify and correct them. The user must keep in mind that the error conditions and messages presented below apply to the IBM/360 and were taken from [2] which is the only up-to-date source of information on the subject.

Before any attempt is made to correct an eventual problem, the errors must be 'exposed'. This very important step is too often jumped over, the user opting to guess directly what went wrong. In order to save time and effort, one should proceed more logically. The user should always check the linkage editor and job scheduler output to ascertain that the proper actions indeed did take place. Any messages such as '-Step-Go-Was Not Run Because of Condition Codes' clearly indicate what operations were not carried out and direct the user to the problem. Using these makes it much easier for the programmer to pinpoint the malfunction and take the appropriate action. If no faulty indications appear in the messages output by the job scheduler (IEF type messages), the linkage editor (IEW type messages), the program producing (IEY) or the object program (IHC) and the results obtained are still suspected to be erroneous, the user then knows he should devote his attention to the mathematics of the problem and revise

the input data (i.e., the output obtained is not the result of a 'computer error').

Some of the possible linkage editor, object program and program producing messages are listed below. These should give the programmer a good idea of what to expect and how to proceed. Experience has shown that even if only a minimum of information is provided, the user greatly benefits from having these simple explanations at hand.

1. IEW000 (control statement only)

This message enumerates all the control statements passed to the linkage editor. INCLUDE and ENTRY cards are listed for reference. It is not an error message.

2. IEW0132 ERROR - SYMBOL PRINTED IN AN UNRESOLVED EXTERNAL REFERENCE

This indicates that the symbol printed to the right of IEW0132 is a subprogram or subroutine which was not in the specified load module library or other modules passed to the linkage editor for processing. The user must make sure the correct subroutine library was specified (i.e., LINCON or LINCONF as required for proper mode of operation), and that the subroutine name requested was correctly spelled.

3. IEW0222 ERROR - CARD PRINTED CONTAINS INVALID INPUT FROM OBJECT MODULE.

In this case, either some control cards were missing, thus causing the editor to interpret wrongly the cards that followed, or some of the cards were punched incorrectly. The deck should be checked.

4. IEW0342 - LIBRARY SPECIFIED DOES NOT CONTAIN MODULE.

The subprogram or subroutine name specified on the INCLUDE control card was not found in the LINCON library. The user must make sure the INCLUDE card was punched as follows:

INCLUDE SYSLIB(member)

where 'member' is the appropriate subprogram name.

5. IHC900I EXECUTION TERMINATING DUE TO ERROR COUNT FOR ERROR NUMBER 217

IHC217I FIFOS - END OF DATA SET ON UNIT 5

Here the computer stopped executing due to lack of data. At that instant, the problem might have been completely solved or not. It is advisable not to take any chances. Again the data deck should be thoroughly checked to ascertain that the cards were set up properly and the data deck incorporated was really the one for the specified subroutine.

6. IHC215 CONVERT - ILLEGAL DECIMAL CHARACTER (decimal character)

The computer found the given decimal character where a number was expected. Either the data cards were improperly set up, the subprogram name specified was incorrect or the FORTRAN format specified was not adhered to. Remedial actions should be taken accordingly.

7. IEY032I NUL PROGRAM

This message indicates that no exterior subroutine was provided when needed and that the computer considered all

the data expected from this subroutine to be zero. Even if this situation can sometimes be used to advantage, it is not recommended here. The programmer should incorporate all required subroutines in his deck. Note that this error can only occur while accessing the library under Mode Two.

8. No error condition messages printed out but incomplete or no results were output by the computer. Here many things could have gone wrong, but most likely one of the following occurred:

- While operating under Mode Two, the ENTRY card was not provided where required. The user must verify the program cards for correctness.

- While operating under Mode Three, insufficient region size was specified. The remedial action is then to increase region size.

- While operating under any of the three modes and the two conditions described above were not the cause, insufficient running time was allocated for the program. If the CPU time indicated on the output and the one specified on the JOB card matched, the user should then allow more time for computation.

- If none of the above, an underflow or overflow condition may have occurred, causing the program to stop. In this case the linkage editor and job scheduler output will indicate a completion code - OCF. The user must verify the data cards and make the appropriate corrections.

The error conditions listed above are obviously not the only ones that can occur, but they are the ones a user is most likely to come across while employing the subroutine library called LINCON.

III. THE LINEAR CONTROL PROGRAMS

A. INTRODUCTION

The previous section dealt with the control statements and the card deck arrangements required to introduce the computer jobs to the operating system and tell the latter everything it needs to know about the input and output requirements. This chapter introduces the theory necessary to use the programs, presents a precise description of all subroutines and data cards and gives detailed examples taken among problems that were solved on the computer.

1. Outline

The subprograms are divided into three classes: the transfer function subprograms, the time response and matrix manipulation subprograms and the modern control subprograms. The first set allows the user to obtain a root locus starting from a block diagram or signal flow graph (RTLOC), the roots of a polynomial and their locus (PRTLOC), the Bode and Nyquist frequency plots (FRESP), the partial fraction expansion of the ratio of two polynomials (PRFEXP) and, finally, the roots of any polynomial (ROOTS). The second group is composed of three subprograms which are provided for determining the rational time response (RTRESP) and the graphical time response (GTRESP) of linear feedback control systems and for computing the determinant, inverse, characteristic polynomial, eigenvalues, state transition matrix and

the resolvent matrix (BASMAT). The last group of subprograms deals with optimal control design. It permits the user to find the observability index of a control system (OBSERV), to test for both controllability and observability (CONOBS), to obtain the state variable feedback given some performance criterion (STVAR), to determine the complete sensitivity analysis of the closed-loop system poles variation as certain parameters change (SENSIT), to design Luenburger observers (LUEN) and serial compensators (SERCOM), to minimize a performance index when some state variables are inaccessible, to solve the Riccati equation to derive optimal control parameters and continuous Kalman filters (RICATI), to compute the gains of discrete Kalman filters (KALMAN), to evaluate the feedback control gains for discrete linear regulator problems, and, finally, to decouple multiple-input multiple-output systems (MIMO). Table I conveniently summarizes the above.

The purpose of each subprogram and a brief discussion of the theory behind it are given in the subprograms presentation.

2. Input Format

The input format for each of the subprograms is completely described with their presentation and must be referred to in each case. However, since the same general input format is used for all the programs, it is appropriate to point out some of the similarities and the conventions adopted. For instance, to make it easier to remember, most of the groups of data cards have the same arrangement.

TABLE I - The Linear Control Subprograms

Name	Purpose	Mode of Operation	Class
RTLOC	To plot the root of a polynomial equation starting from a feed-back control system block diagram.	One or Three	B
PRTLOC	To plot the root locus of a characteristic polynomial.	Two	B
FRESP	To obtain and plot the frequency response of a rational transfer function over a specified range of frequencies. Both Bode and Nyquist diagrams can be plotted.	One or Three	A/B
PRFEXP	To perform the partial fraction expansion of a rational function.	One or Three	A
ROOTS	To find the roots of a polynomial of order less than or equal to twenty.	One or Three	A
BASMAT	To compute the determinant, the inverse, the characteristic polynomial, the eigenvalues, the state transition matrix, and the resolvent matrix from a given matrix $\underset{\sim}{A}$ (NxN).	One or Three	A
RTRESP	To determine the rational time response of a system (in closed-form).	One or Three	A

TABLE I (Continued)

Name	Purpose	Mode of Operation	Class
GTRESP	To obtain the graphical time response of a system for a specified input.	One or Three	A/B
OBSERV	To determine the observability index for a linear system.	One or Three	A
CONOBS	To check for both observability and controllability of a linear system.	One or Three	A
SENSIT	To study the closed-loop poles variation of a linear feedback system.	One or Three	A/B
STVAR	To calculate the controller gain and feedback coefficients to achieve a desired closed-loop transfer function. Also computes the plant transfer function, internal transfer functions and determines $H_{eq}(s)$, the equivalent single-feedback element.	One or Three	A
LUEN	To design Luenberger Observers to achieve a desired closed-loop transfer function.	One or Three	A
SERCOM	To design a series compensator to achieve a desired closed-loop transfer function.	One or Three	A

TABLE I (Continued)

Name	Purpose	Mode of Operation	Class
RICATI	To solve the differential Riccati equation to determine the optimum control gains for state-regulator problems and/or the continuous Kalman filter gains.	One or Three	A/B
KALMAN	To determine the discrete Kalman filter gains.	Two	A
STREG	To evaluate the discrete feedback gains of linear regulator problems.	One or Three	A
MIMO	To decouple an Nth order system with M inputs and M outputs and place the closed-loop poles of each decoupled subsystem at specified locations.	One or Three	A

a. First Data Card

The purpose of the first data card of any of the subprograms is to identify the problem for future reference and for output data. A maximum of twenty alpha-numeric characters (except $) can be used, starting in column one (format 5A4). On this first card, the user also normally defines the system order and the dimensions of the various matrices (format I2 for each number to be entered). Note that the dollar sign $ has been defined as a STOP and must never be used as problem identification.

b. Matrices

Matrices are entered one row at a time either in their original form or transposed, as specified. The input format table presented with each subprogram indicates the correct form to use. The vectors are always defined using lower case letters while other matrices are identified with capital letters. The matrix elements are punched in ten-column fields (format 8E10 or 8F10), thus a maximum of eight numbers can be given per card. If the order of the system is greater than eight, two cards are needed for every row.

An example will demonstrate the procedure. Assume that the $\underset{\sim}{A}$ and $\underset{\sim}{b}$ matrices are:

$$\underset{\sim}{A} = \begin{bmatrix} 3.19 & 0.00 & -10.11 \\ 2.45 & 6.40 & -0.50 \\ 1.00 & -9.14 & 6.75 \end{bmatrix} \qquad \underset{\sim}{b} = \begin{bmatrix} 1.0 \\ 0.0 \\ 15.2 \end{bmatrix}$$

The given $\underset{\sim}{A}$ and $\underset{\sim}{b}$ matrices are entered using an 8F10.3 format as follows:

card columns	1	11	21
	3.19	0.0	-10.11
$\underset{\sim}{A}$	2.45	6.4	- 0.5
	1.0	-9.14	6.75
$\underset{\sim}{b}^T$	1.0	0.0	15.2

c. Polynomials

The polynomial data can be entered in two different formats referred to as P mode (polynomial form) and F mode (factored form). If P mode is selected, the letter P (format A1) followed by the degree of the polynomial (format I2) are entered on one card. The coefficients of the polynomial are placed on the next card(s) each in ten column fields (format 8F10 or 8E10). The polynomials are always presented in ascending order, the constant term given first and the coefficient of the highest term assumed to be unity. In other words, the last coefficient entered will always be interpreted as being 1.0, thus can be entered either as '1.0' or as a blank. Again an example best illustrates the principles.

The given four polynomials are entered using an 8F10.3 format:

(1) Polynomials:

(i) $2 + 4s + s^2$

(ii) $s + 5s^2 + 6s^3 + s^4$

(iii) 1. (highest degree coefficient of a zero order polynomial)

(iv) $4 + s^2 + s^4 + 3s^6 + s^8$

(2) Computer data cards:

card columns	1	11	21	31	41	51	61	71
	P02							
	2.0	4.0	1.0					
	P04							
	0.0	1.0	5.0	6.0	1.0			
	P00							
	1.0							
	P08							
	4.0	0.0	1.0	0.0	1.0	0.0	3.0	0.0
	1.0							

If it is desired to enter the polynomial in factored form, then the F mode is chosen. This choice is indicated by placing the letter F (format A1) in the first column followed by the degree of the polynomial in the next two (format I2). The factors are then entered one per card, the real part in the first ten column field and the imaginary part in the next ten columns (format 2E10 or 2F10). An unusual convention was picked to enter all the possible factors. The user must be careful and make sure his notation agrees with the following:

(1) The real part of the root is entered as <u>positive</u> if the factor is in the <u>left half plane</u>.

(2) The real part of the root is entered as <u>negative</u> if the factor is in the <u>right half plane</u>.

(3) <u>Only one</u> of the complex conjugate roots is entered and it must be with the one with the <u>positive</u> imaginary part.

(4) If the polynomial is a constant, it must equal 1.0 and be entered in the P mode as shown below. Examples covering many possibilities are shown next.

Factored polynomials:

(i) (s + 3) (s - 1)

(ii) s(s + 4)(s + 1 + j)(s + 1 - j)

(iii) 1.0

(iv) (s - 1)(s - 2 + j5)(s - 2 - j5)(s + 3)(s + 3)

Computer data cards:

```
        F02
i)      3.
        -1.
        F04
        0.0
ii)     4.0
        1.0      1.0

         P00
iii)     1.0
```

F04

-1.0

(iv) -2.0 5.0

3.0

3.0

One good way to remember how to work around this confusing notation is to always enter the real parts as they appear in the factored polynomial and include the positive imaginary part only. In other words, one can analyse any situation in the following manner:

$$(s + 0)\ (s + 3)\ (s - 1)\ (s + 1 + j2)\ (s + 1 - j2)$$

where the circles indicate the numbers to be punched.

d. Multiple Runs

One last common characteristic of the input data is that one or several data decks pertaining to the same subprogram can be stacked and run as a single job. In other words, one complete data deck is prepared for each problem but the decks are all put one on top of the other and read in to the computer preceded only by one set of control cards.

The user must realize,however,that this feature implies more runs to be performed in a single job and thus the time limit to be specified on the JOB control card must be estimated accordingly.

3. Output Format

The output of all of the subprograms is quite comprehensible and need not be explained. Nevertheless confusion may arise due to certain factors that are commented upon here. For the matrices, the same rules as for the input apply; vectors are listed out as transpose matrices and all other types of matrices are presented one row at a time. For convenience, polynomials are always output both in polynomial and factored forms no matter how they were provided as input. As for the input, the coefficients appear in ascending order, the constant term first. In factored form, the roots are listed with their normal sign convention; the left half plane roots are negative and those in the right half plane positive.

Hence there is a sign inversion between the input and the output for the factored case.

B. THE TRANSFER FUNCTION SUBPROGRAMS

This set helps the user to analyse or design feedback control systems by providing a means of obtaining quickly the roots locus, Bode diagrams, Nyquist plots, partial fraction expansions and polynomial roots.

1. Root Locus (RTLOC)

This subprogram calculates and plots the roots of the equation

$$1 + K\,G(s) = 0$$

where G(s) is a rational function of the form

$$G(s) = \frac{N(s)}{D(s)}$$

The user must provide N(s), D(s) and a range of value for K. Since a choice of two ways to vary K from minimum to maximum gain exists, an option card is also required.

a. Input

The observations and the table presented below should be sufficient to use the subprogram which can be called under Mode One or Mode Three (as described in Chapter II):

(1) N(s) can be input either in P form or F form

(2) D(s) can be input either in P form or F form

(3) K values must be all positive or all negative. If both are desired, two separate runs must be made. Also, the maximum gain value cannot be zero.

(4) An option card must be included to indicate whether or not a particular region of the root locus is to be drawn (zoom capability). A blank option card implies no option is desired. Note that selecting a specific region improves the accuracy of the plot.

The last card tells the computer to plot only the roots locus in the rectangle in the s plane defined by:

$$\sigma_{min} \leq Re[s] \leq \sigma_{max}$$

$$\omega_{min} \leq Im[s] \leq \omega_{max}$$

as illustrated in Figure 3-1.

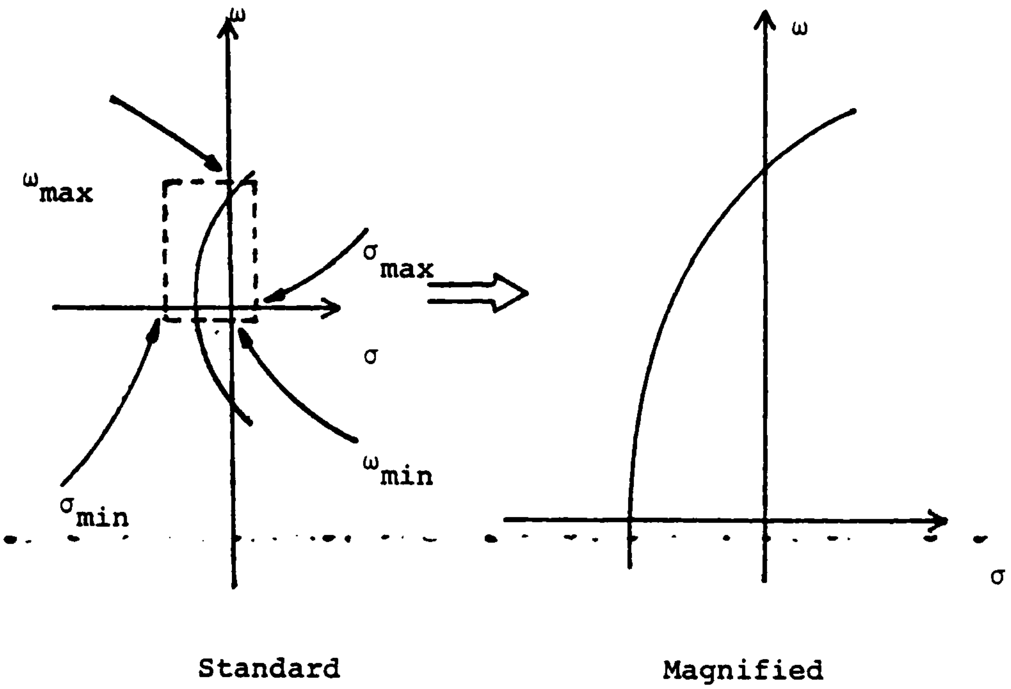


Fig 3-1 Magnified and Standard Root Locus

Thus a standard root locus plot is obtained by leaving the option card blank, while a magnified root locus is plotted by punching a "1" in the first column and specifying the minimum and maximum values of σ and ω. The input formats for RTLOC are given in Table II.

b. Output

The problem identification is given, followed by the numerator and denominator polynomials, both in factored and 'ascending coefficients' form, and the minimum and maximum

Entry	Input Description	Format	Columns Used
1	Problem Identification	5A4	1-20
2	letter P or F (for P form and F form), Order of N(s) ($\leq$ 10)	A1, I2	1, 2-3
3	Enter N(s) in format specified on previous card	8E10.0	1-10, 11-20, etc.
4	Letter P or F (for P form and F form), Order of D(s) ($\leq$ 10)	A1, I2	1, 2-3
5	Enter D(s) in format specified on previous card	8E10.0	1-10, 11-20, etc.
6	Minimum value of gain, maximum value of gain ($\neq$ 0)	8E10.0	1-10, 11-20
7	No option = blank card Option $\neq$ 0 minimum value of σ, maximum value of σ, minimum value of ω, maximum value of ω	I1, 9X, 8E10.0	1, 11-20, 21-30, 31-40, 41-50

Table II - Input Format Table for RTLOC

gains. The roots' real and imaginary parts are then listed as the gain varies from its minimum to maximum value. Finally the root locus plot is printed out. Note that the graph produced has square grids so that the true angles can be measured.

This is normally a class B program and time = 2 should be specified on the JOB card.

c. Example

Obtain the root locus of the following feedback control system:

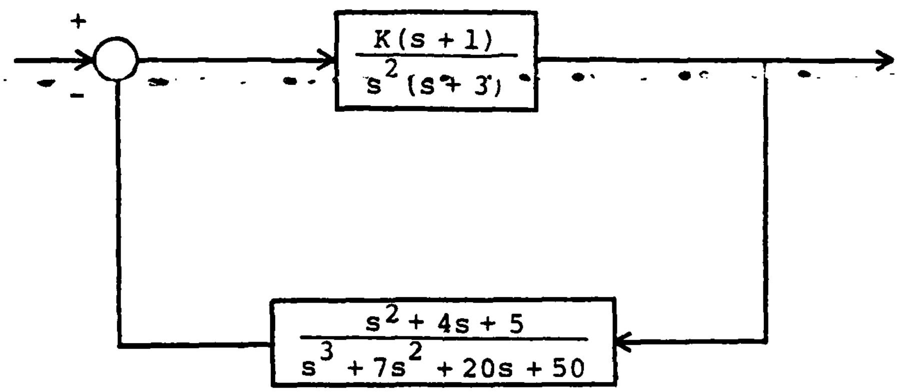


Fig 3-2 Feedback Control System for RTLOC Test.

The equation for which the roots are to be found is then:

$$1 + K \frac{(s+1)(s^2+4s+5)}{s^2(s+3)(s^3+7s^2+20s+50)} = 0$$

It agrees with the RTLOC structure so one can proceed further.

$$N(s) = (s+1)(s+2+j1)(s+2-j1)$$

and

$$D(s) = 0 + 0s + 150s^2 + 110s^3 + 41s^4 + 10s^5 + s^6$$

Here it is easier to enter N(s) in factored form and D(s) as an ascending polynomial.

As a first guess, the range of variation of the gain is chosen to be from 0.0 to 100.0 and since the expected plot is unknown, no option is taken.

This completes the work. The computer does the rest provided the cards are punched as follows:

```
// (standard OS JOB card),TIME=2
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(RTLOC)
/*
//GO.SYSIN^DD^*
RTLOC TEST
F03
1.
2.    1.
P06
0.0   0.0   150.   110.   41.   10.
```

```
0.0    100.
(blank card)
/*
```

The results appear in Figs. 3-3A and 3-3B. Note that the user should mark the open-loop poles and zeroes for easier interpretation.

2. Root Locus (PRTLOC)

As the name indicates, this subprogram is a modified version of RTLOC. It calculates the roots of a polynomial and plots them. The method to input the data differs slightly but the ultimate goal remains the same.

a. Input

This subprogram can only be used under Mode Two of operation. The coefficients of the polynomial must be entered using a simple subroutine called RPOL(C,G) which must be typed as follows:

```
SUBROUTINE RPOL(C,G)
DIMENSION C(20)
C(1) = fnct (G)
C(2) = fnct (G)
  .   .   .
  .   .   .
  .   .   .
C(n+1) = 1.0
RETURN
END
```

where n = order of the equation.

```
ROOT LOCUS PROGRAM
PROBLEM IDENTIFICATION - RTLCC TEST
******************************************

NUMERATOR COEFFICIENTS IN ASCENDING POWERS OF S

     5.000     9.000     5.000     1.000

OPEN-LOOP ZEROES
  REAL PART  IMAG. PART
 -2.000E 00 -1.000E 00
 -2.000E 00  1.000E 00
 -1.000E 00  0.0

DENOMINATOR COEFFICIENTS IN ASCENDING POWERS OF S

     0.0       0.0     150.000   110.000    41.000    10.000     1.000

OPEN-LOOP POLES
  REAL PART  IMAG. PART
 -1.000E 00 -3.000E 00
 -1.000E 00  3.000E 00
 -5.000E 00  0.0
 -3.000E 00  0.0
  0.0        0.0
  0.0        0.0

MIN. GAIN =   0.0                 MAX. GAIN =   1.00E 02
******************************************

  1       GAIN =  0.0

        ROOTS ARE
  REAL PART  IMAG. PART

 -1.000E 00 -3.000E 00
 -1.000E 00  3.000E 00
 -5.000E 00  0.0
 -3.000E 00  0.0
  0.0        0.0
  0.0        0.0

  2       GAIN =  5.750E-02

        ROOTS ARE
  REAL PART  IMAG. PART

 -9.985E-01 -3.000E 00
 -9.985E-01  3.000E 00
 -5.002E 00  0.0
 -2.999E 00  0.0
 -1.023E-03 -4.377E-02
 -1.023E-03  4.377E-02

  3       GAIN =  1.236E-01

        ROOTS ARE
  REAL PART  IMAG. PART

 -2.998E 00  0.0
 -9.969E-01 -3.001E 00
 -9.969E-01  3.001E 00
 -5.004E 00  0.0
 -2.199E-03 -6.416E-02
 -2.199E-03  6.416E-02

  4       GAIN =  1.997E-01

 39       GAIN =  7.726E 01

        ROOTS ARE
  REAL PART  IMAG. PART

 -6.539E 00  0.0
  4.425E-01 -3.961E 00
  4.425E-01  3.961E 00
 -2.298E 00  0.0
 -1.029E 00 -7.531E-01
 -1.029E 00  7.531E-01

 40       GAIN =  8.890E 01

        ROOTS ARE
  REAL PART  IMAG. PART

  5.578E-01 -4.108E 00
  5.578E-01  4.108E 00
 -6.695E 00  0.0
 -2.214E 00  0.0
 -1.103E 00 -7.277E-01
 -1.103E 00  7.277E-01
```

Figure 3-3A Root Locus Test - Numerical Output

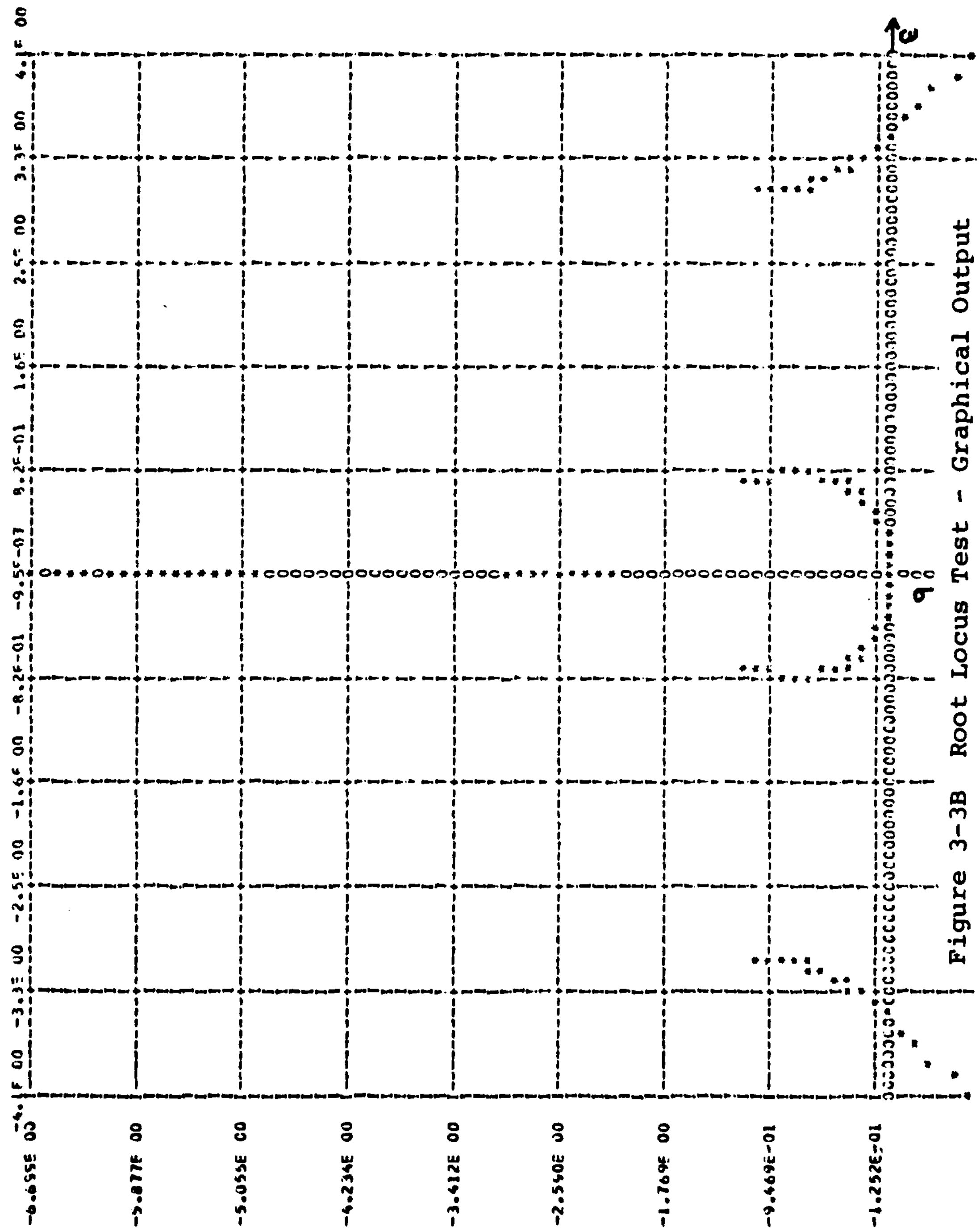


Figure 3-3B Root Locus Test - Graphical Output

C(1),...C(n+1)	=	coefficients of the polynomial, in ascending order. Note that the coefficient of the highest order term C(n+1) must be 1.0 and need not be entered.
fnct (G)	=	defining coefficient equation in terms of G, the gain. The function could very well be a constant only.

The remaining data, i.e., the problem identification, range of gain values and option,are entered as follows:

ENTRY	Input Description	Format	Columns Used
1	Problem identification,	5A4,	1 - 20
	Order of the polynomial,	I2	21 - 22
2	minimum value of gain,	8E10.0	1 - 10, 11 - 20
	maximum value of gain ($\neq$ 0)		
3	{no option = blank card}	I1, 9x,	1, 11-20, 21-30, 31-40
	option $\neq$ 0,	8E10.0	41-50
	minimum value of σ,		
	maximum value of σ,		
	minimum value of ω,		
	maximum value of ω.		

Table III - Input Format Table for PRTLOC

Here again, the gain values must be either all positive or all negative and the maximum gain cannot equal zero.

The first card, in addition to the usual problem identification must contain the polynomial order in columns 21-22.

The last card is used to indicate whether or not a portion only of the root locus is to be refined and plotted. If the option is selected, a number greater than zero is punched in the first column, followed in columns 11-50 by the parameters defining the rectangular portion to be blown up (see example). If this option is not desired, the card is left blank. Note that this version permits us to find the roots of any characteristic equation with a single varying parameter G.

b. Output

The problem identification and the minimum and the maximum gain are first listed out for future reference. Next, the root values are given as the gain varies and the root locus plotted. The execution time to be included on the JOB card should be "time = 2".

c. Example

While trying to solve problem 7.26 in Shinners [3], one comes across the following characteristic equation for part of the system:

$$s^4 + 9.15s^3 + (1.32 + 20K_2)s^2 + (26K_2 - .15)s + (6K_2 + 0.675) = 0$$

At this point the root locus is desired to determine what value of K_2 is required to satisfy some criterion. Since

the characteristic equation is specified explicitly, PRTLOC is selected.

First the coefficients are sorted out and written as functions of G where G is equal to K_2.

s**0 coefficient : C(1) = 0.675 + 6.*G

s**1 coefficient : C(2) = -0.15 + 26.*G

s**2 coefficient : C(3) = 1.32 + 20.*G

s**3 coefficient : C(4) = 9.15

Note that the coefficient of the highest order term is always taken as 1.0 and need not be included. The above data is to be entered by writing the subroutine RPOL(C,G).

The order of the equation is 04. The range of gain values to be investigated is from 0.0 to 20.0 and since no refined plot is desired at this point the last card is a blank card.

The following cards then constitute the entire deck to be input to the computer:

```
// (standard OS JOB card),TIME=2
//⌃EXEC⌃LINCONF
//FORT.SYSIN⌃DD⌃*
      SUBROUTINE RPOL(C,G)
      DIMENSION C(20)
      C(1)=0.675+6.0*G
      C(2)=-0.15+26.0*G
      C(3)=1.32+20.*G
      C(4)=9.15
      RETURN
      END
```

```
/*
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(PRTLOC)
^^ENTRY^PRTLOC
/*
//GO.SYSYN^DD^*
PRTLOC TEST ONE    04
0.0    20.0
(blank card)
/*
```

The results obtained with this first run as are shown in Figs. 3-4A and 3-4B. However they do not permit us to evaluate the gain precisely enough and a second run is made, this time using the option. The rectangular portion where magnification is desired is defined by:

$$\sigma_{min} = -5.$$

$$\sigma_{max} = 5.$$

$$\omega_{min} = -1.$$

$$\omega_{max} = 5.$$

Note that this option not only concerns the plotting but also produces a larger number of gain values. Thus, in order not to have too many values listed out unnecessarily, it is good practice to re-specify the range. It was decided to change it to vary from 0.0 to 10.0.

```
ROOT LOCUS PROGRAM
PROBLEM IDENTIFICATION - PRTLOC TEST ONE
*********************************************
MIN. GAIN =   0.0              MAX. GAIN =   2.00E 01
*********************************************
  1      GAIN =  0.0

       ROOTS ARE
 REAL PART   IMAG. PART

-9.001E 00  0.0
 1.752E-01 -3.455E-01
 1.752E-01  3.455E-01
-4.997E-01  0.0
  2      GAIN =  5.750E-02

       ROOTS ARE
 REAL PART   IMAG. PART

-8.888E 00  0.0
 1.072E-01 -4.787E-01
 1.072E-01  4.787E-01
-4.768E-01  0.0
  3      GAIN =  1.236E-01

       ROOTS ARE
 REAL PART   IMAG. PART

-8.755E 00  0.0
 2.868E-02 -5.972E-01
 2.868E-02  5.972E-01
-4.527E-01  0.0
  4      GAIN =  1.997E-01

       ROOTS ARE
 REAL PART   IMAG. PART

-8.597E 00  0.0
-6.193E-02 -7.100E-01
-6.193E-02  7.100E-01
-4.283E-01  0.0
  5      GAIN =  2.871E-01

       ROOTS ARE
 REAL PART   IMAG. PART

-8.410E 00  0.0
-1.666E-01 -8.201E-01
-1.666E-01  8.201E-01
-4.032E-01  0.0
  6      GAIN =  3.877E-01

       ROOTS ARE
 REAL PART   IMAG. PART

-8.184E 00  0.0
-2.888E-01 -9.278E-01
-2.888E-01  9.278E-01
-3.884E-01  0.0
  7      GAIN =  5.033E-01

       ROOTS ARE
 REAL PART   IMAG. PART

-7.910E 00  0.0
-4.337E-01 -1.032E 00
-4.337E-01  1.032E 00
-3.728E-01  0.0

 28      GAIN =  1.630E 01

       ROOTS ARE
 REAL PART   IMAG. PART

-3.910E 00 -1.736E 01
-3.910E 00  1.736E 01
-1.028E 00  0.0
-3.026E-01  0.0
 29      GAIN =  1.881E 01

       ROOTS ARE
 REAL PART   IMAG. PART

-1.024E 00  0.0
-3.912E 00 -1.875E 01
-3.912E 00  1.875E 01
-3.023E-01  0.0
```

Figure 3-4A PRTLOC Test One - Numerical Output

ROOT LOCUS PROGRAM
PROBLEM IDENTIFICATION - PRTLOC TEST ONE

MIN. GAIN = 0.0 MAX. GAIN = 2.00E 01

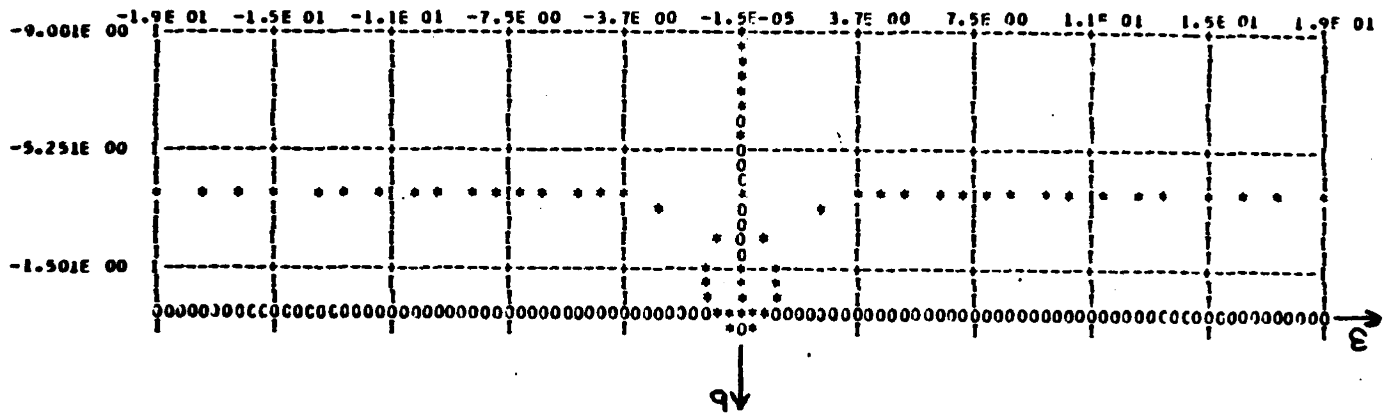


Figure 3-4B PRTLOC Standard Root Locus Plot

Since the characteristic equation did not change, only the data deck is to be modified. These last three cards are given below.

```
PRTLOC,TEST TWO  04
0.0     10.0
1       -5.0       5.0      -1.0      5.0
```

This magnified portion of the root locus is presented in Fig. 3-5A and 3-5B

3. Frequency Response (FRESP)

This subprogram determines the frequency response of a rational transfer function

$$G(s) = K \frac{N(s)}{D(s)}$$

and plots the response in the form of a Bode or/and Nyquist diagram, as specified.

a. Input

The problem identification, the gain and the two polynomials N(s) and D(s) are entered followed by the minimum and the maximum radian frequency values, the number of frequency values to be used (smaller or equal to 500), the interpolation and discrete value options, the Bode plots and the Nyquist diagrams options and, only if required, the discrete frequency values.

It might look complex at first, but the subprogram is very simple to use and the results obtained are quite good. The routine is accessed under Mode One or Mode Three. The

```
ROOT LOCUS PROGRAM
PROBLEM IDENTIFICATION - PRTLOC TEST TWO

**********************************************

MIN. GAIN =   0.0                 MAX. GAIN =   1.00E 01

**********************************************

OPTION HAS BEEN TAKEN

SIGMA MIN =  -5.00E 00      SIGMA MAX =   5.00E 00
OMEGA MIN =  -1.00E 00      OMEGA MAX =   5.00E 00

**********************************************

    1        GAIN =  0.0

          ROOTS ARE
    REAL PART   IMAG. PART

   -9.001E 00  0.0
    1.752E-01 -3.455E-01
    1.752E-01  3.455E-01
   -4.997E-01  0.0

    2        GAIN =  2.080E-02

          ROOTS ARE
    REAL PART   IMAG. PART

   -8.960E 00  0.0
    1.507E-01 -3.987E-01
    1.507E-01  3.987E-01
   -4.913E-01  0.0

    3        GAIN =  4.243E-02

          ROOTS ARE
    REAL PART   IMAG. PART

   -8.917E 00  0.0
    1.251E-01 -4.476E-01
    1.251E-01  4.476E-01
   -4.827E-01  0.0

    4        GAIN =  6.493E-02

          ROOTS ARE
    REAL PART   IMAG. PART

   -8.873E 00  0.0
    9.843E-02 -4.934E-01
    9.843E-02  4.934E-01
   -4.740E-01  0.0

    5        GAIN =  8.833E-02

          ROOTS ARE
    REAL PART   IMAG. PART

   -8.826E 00  0.0
    7.065E-02 -5.371E-01
    7.065E-02  5.371E-01
   -4.652E-01  0.0

    6        GAIN =  1.127E-01

          ROOTS ARE
    REAL PART   IMAG. PART

   -8.777E 00  0.0
    4.172E-02 -5.792E-01
    4.172E-02  5.792E-01
   -4.565E-01  0.0

    76       GAIN =  9.331E 00

          ROOTS ARE
    REAL PART   IMAG. PART

   -3.897E 00 -1.272E 01
   -3.897E 00  1.272E 01
   -1.051E 00  0.0
   -3.046E-01  0.0

    77       GAIN =  9.725E 00

          ROOTS ARE
    REAL PART   IMAG. PART

   -3.899E 00 -1.303E 01
   -3.899E 00  1.303E 01
   -1.048E 00  0.0
   -3.044E-01  0.0
```

Figure 3-5A PRTLOC Test Two - Numerical Output

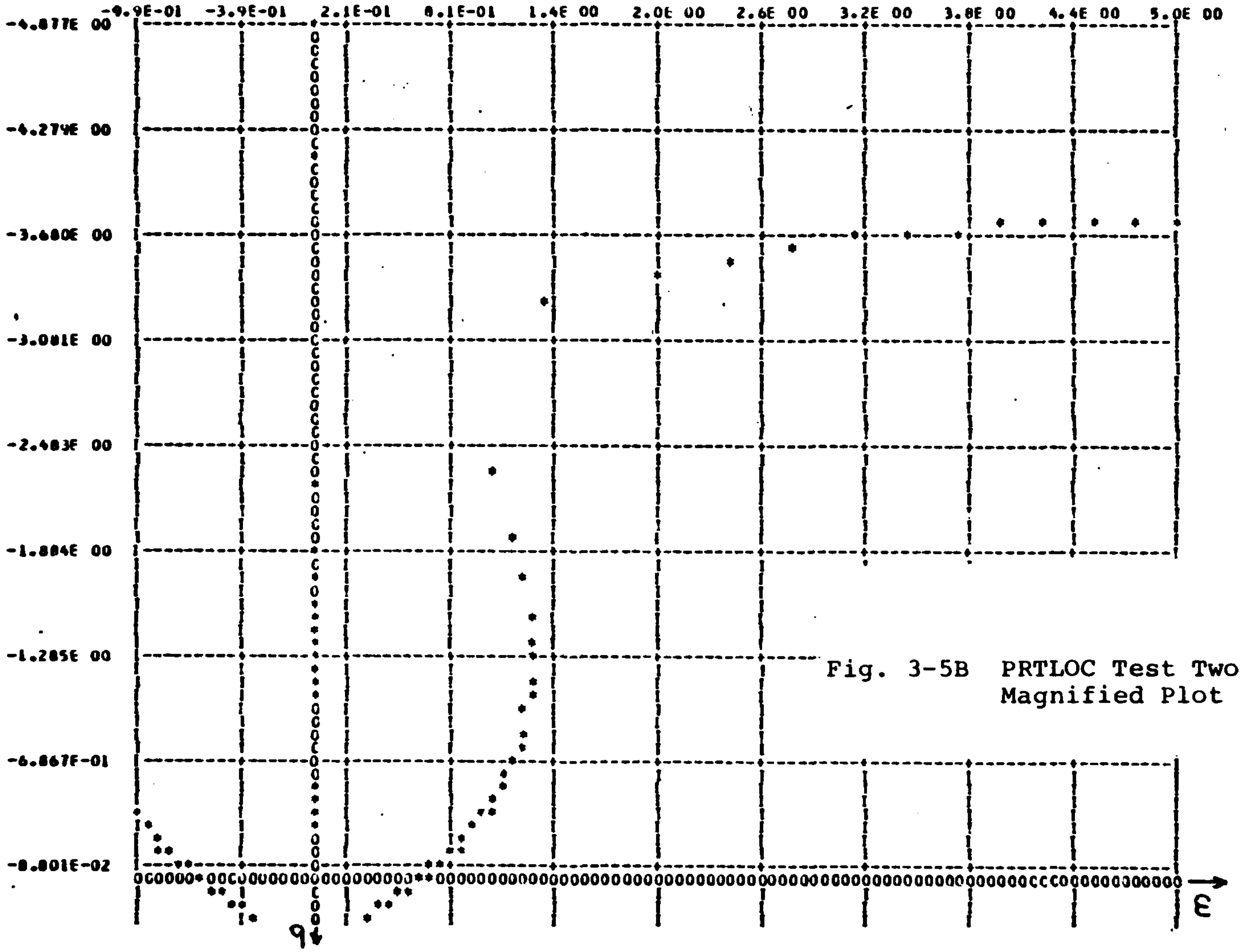


Fig. 3-5B PRTLOC Test Two - Magnified Plot

input format table and the example that follows demonstrate the procedure to be used.

Entry	Input Description	Format	Columns Used
1	Problem identification	5A4	1-10
2	The gain K	8E10.0	1-10
3	letter P or F (for P form or F form), order of N(s) $\leq$ 10	A1,I2	1, 2-3
4	Enter N(s) in format specified on the previous card	8E10.0	1-10, 11-20, etc.
5	letter P or F (for P form or F form), order of D(s) $\leq$ 10	A1, I2	1, 2-3
6	enter D(s) in format specified on the previous card	8E10.0	1-10, 11-20, etc.
7	minimum radian frequency ($\neq$0), maximum radian frequency,	2E10.0	1-10, 11-20,
	number of frequency values to be used ($\leq$500)	I3	21-23,
	option I: logarithmic interpolation = 000 discrete values supplied = 001 linear interpolation = 002	I3	24-26,
	option B: Bode plot = 000 no Bode plot = 001	I3	27-29
	option N: Nyquist plot = 000 no Nyquist plot = 001	I3	30-32
8 (if and only if option I = 001	discrete frequency values	8E10.0	1-10, 11-20, etc.

Table IV - Input Format Table for FRESP

Here the option card is a bit complex, but it provides great flexibility. The following ideas should help clarify the concept:

(1) The first three entries specify the range and the number of data points for the Bode and/or Nyquist plot. One must recall that the Bode magnitude plot is log-log; the Bode phase plot is log-linear (angles in degree) while the Nyquist is a polar plot. Thus, minimum and maximum radian frequency values should be carefully chosen. For example, $\omega_{min} = 0.01$ and $\omega_{max} = 100$ could be a good choice in a given problem while being absurd for another one.

(2) Option I specifies the type of interpolation to be used to generate the values between the minimum and maximum frequency.

If Option I = 000, logarithmic interpolation is used to select the frequency value. Either plot can be obtained while specifying this option.

If Option I = 001, the user must enter on the following cards the frequency values for which he wants $G(j\omega)$ to be evaluated. The number of frequency values must again be less or equal to 500. No plot can be obtained when this option is selected, only tabular outputs.

If Option I = 002, linear interpolation is used to select the frequency values for which $G(j\omega)$ is to be computed. Only the Nyquist plot can be obtained when this option is used.

(3) Option B indicates whether or not Bode diagrams are to be drawn.

If Option B = 000, Bode plots will be output

If Option B = 001, Bode plots will not be output

(4) Similarly, option N is used to specify whether a Nyquist plot is desired or not.

If option N = 000, it is desired

If option N = 001, it is not desired.

(5) The options card is not followed by any card except when option I is equal to 001. If this is the case, the frequency values must be entered using an 8E10.0 format. Note that an option card containing only the minimum and the maximum frequency values and the number of points to be evaluated indicates that both Bode and Nyquist plot are desired.

b. Output

The problem identification, the value of the gain, the coefficients of the polynomials N(s) and D(s) as well as their roots are listed for reference. Next, the radian frequency, the real and imaginary part of $G(j\omega)$, the magnitude $|G(j\omega)|$, the magnitude in db, the phase in radians and the phase in degrees are printed out in tabular form for the indicated number of frequency values (smaller or equal to 500).

If option B = 000 has been selected, the magnitude and phase Bode diagrams are given. <u>Note that the phase angles are always normalized to lie between -180° and +180°</u>.

If option N has been requested, the Nyquist diagram is plotted with the points linearly, or logarithmically, spaced out.

Normally the CPU time required to run the program is less than 20 seconds (class A).

c. Example One

The Bode plot for the loop transfer function is to be obtained for the following system:

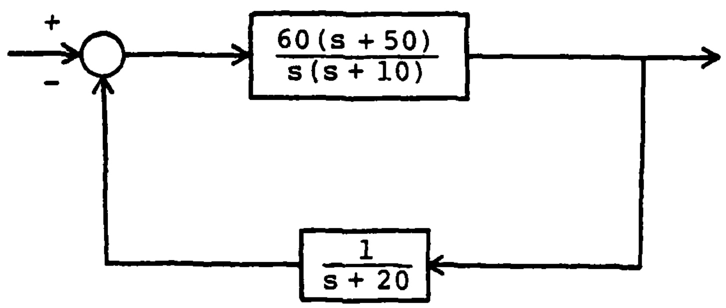


Fig. 3-6 Compensated Control System for FRESP Test

The first step is to define G(s). Here it is simply

$$G(s) \;=\; K\,\frac{N(s)}{D(s)} \;=\; 60\left[\frac{(s+50)}{s(s+10)(s+20)}\right]$$

The gain is 60 and since both N(s) and D(s) are already factored, they can best be entered using F form. The minimum and maximum frequencies are arbitrarily chosen to be 0.1 and 100, respectively. The number of frequency values for which $G(j\omega)$ is to be evaluated is 50. Since a Bode diagram is desired, option I <u>must</u> equal 000 (logarithmic interpolation)

and option B also equals 000. In this case a Nyquist plot is not desired and option N is entered as 001.

The control cards and data deck to run the sub-program are then:

```
// (standard OS JOB card)
//‸EXEC‸LINCON
//LINK.SYSIN‸DD‸*
‸ INCLUDE‸SYSLIB(FRESP)
/*
//GO.SYSIN DD *
FRESP TEST ONE
60.
F01
50.
F03
0.0
10.0
20.0
0.1    100.     050000000001
/*
```

The Bode diagrams are shown in Fig. 3-7 (A-C). Note that the phase versus frequency diagram presented is not in error but simply due to the fact that the angle values are normalized to be within -180 and +180 degrees. This is useful since it permits one to rapidly determine where the -180° crossing occurs.

```
FREQUENCY RESPONSE
PROBLEM IDENTIFICATION - FRESP TEST ONE
**********************************************

GAIN = 6.000000E 01

NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S

5.000000E 01 1.000000E 00

NUMERATOR ROOTS ARE
   REAL PART        IMAG. PART
-5.000000E 01 0.0

DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S

0.0           2.000000E 02 3.000000E 01 1.000000E 00

DENOMINATOR ROOTS ARE
   REAL PART        IMAG. PART
 0.0            0.0
-1.000000E 01 0.0
-2.000000E 01 0.0
**********************************************
```

RADIAN FREQ.	REAL PART	IMAGINARY PART	MAGNITUDE	MAG IN DB	PHASE (RAD)	PHASE (DEG)
9.999996E-02	-1.549774E 00	-1.499785E 02	1.499911E 02	4.352130E 01	-1.583797E 00	-9.074487E 01
1.151394E-01	-1.549698E 00	-1.302517E 02	1.302665E 02	4.229663E 01	-1.585764E 00	-9.085759E 01
1.325705E-01	-1.549595E 00	-1.131182E 02	1.131350E 02	4.107193E 01	-1.588030E 00	-9.098743E 01
1.526415E-01	-1.549468E 00	-9.823628E 01	9.825546E 01	3.984714E 01	-1.590638E 00	-9.113667E 01
1.757506E-01	-1.549295E 00	-8.531003E 01	8.533234E 01	3.862225E 01	-1.593642E 00	-9.130899E 01
2.023587E-01	-1.549063E 00	-7.408194E 01	7.410742E 01	3.739725E 01	-1.597100E 00	-9.150711E 01
2.329942E-01	-1.548757E 00	-6.432863E 01	6.435815E 01	3.617206E 01	-1.601081E 00	-9.173518E 01
2.682683E-01	-1.548354E 00	-5.585589E 01	5.588915E 01	3.494664E 01	-1.605664E 00	-9.199780E 01
3.088427E-01	-1.547817E 00	-4.849498E 01	4.853410E 01	3.372093E 01	-1.610943E 00	-9.230009E 01
3.556460E-01	-1.547112E 00	-4.209956E 01	4.214435E 01	3.249481E 01	-1.617014E 00	-9.264809E 01
4.094889E-01	-1.546171E 00	-3.654213E 01	3.659392E 01	3.126817E 01	-1.624004E 00	-9.304861E 01
4.714834E-01	-1.544927E 00	-3.171217E 01	3.177176E 01	3.004082E 01	-1.632051E 00	-9.350961E 01
5.428635E-01	-1.543279E 00	-2.751355E 01	2.758209E 01	2.881253E 01	-1.641305E 00	-9.404008E 01
6.250552E-01	-1.541098E 00	-2.386265E 01	2.394151E 01	2.758302E 01	-1.651962E 00	-9.465048E 01
7.196746E-01	-1.538214E 00	-2.068694E 01	2.077754E 01	2.635187E 01	-1.664216E 00	-9.535257E 01
8.286353E-01	-1.534404E 00	-1.792314E 01	1.803772E 01	2.511856E 01	-1.678308E 00	-9.616007E 01
9.540864E-01	-1.529369E 00	-1.551635E 01	1.563595E 01	2.388247E 01	-1.694506E 00	-9.708804E 01
1.098530E 00	-1.522733E 00	-1.341875E 01	1.355541E 01	2.264249E 01	-1.713115E 00	-9.815425E 01
1.048042E 01	-7.536300E-01	-4.175065E-01	8.940946E-01	-9.723112E-01	-2.655743E 00	-1.521624E 02
1.206768E 01	-6.471703E-01	-2.629297E-01	6.985607E-01	-3.115916E 00	-2.755697E 00	-1.578899E 02
1.389460E 01	-5.154305E-01	-1.525197E-01	5.375212E-01	-5.392056E 00	-2.853853E 00	-1.635161E 02
1.599824E 01	-3.558505E-01	-7.840633E-02	4.074653E-01	-7.799186E 00	-2.947959E 00	-1.689056E 02
1.842029E 01	-3.026438E-01	-3.223219E-02	3.045543E-01	-1.032670E 01	-3.035559E 00	-1.739247E 02
2.120901E 01	-2.246757E-01	-6.076008E-03	2.247779E-01	-1.296493E 01	-3.114557E 00	-1.784510E 02
2.441994E 01	-1.640062E-01	6.823257E-03	1.641420E-01	-1.569560E 01	-3.183173E 00	-1.823824E 02
2.811699E 01	-1.113016E-01	1.169598E-02	1.188799E-01	-1.849782E 01	-3.240136E 00	-1.856462E 02
3.237375E 01	-8.474410E-02	1.221983E-02	9.542058E-02	-2.134842E 01	-3.284801E 00	-1.882053E 02
3.727496E 01	-6.054515E-02	1.074339E-02	6.149095E-02	-2.422375E 01	-3.317208E 00	-1.900621E 02
4.291626E 01	-4.329426E-02	8.618068E-03	4.414861E-02	-2.710164E 01	-3.338059E 00	-1.912567E 02
4.941575E 01	-3.107937E-02	6.527890E-03	3.175753E-02	-2.996304E 01	-3.348621E 00	-1.918619E 02
5.689706E 01	-2.242734E-02	4.755989E-03	2.292607E-02	-3.279340E 01	-3.350559E 00	-1.919729E 02
6.551100E 01	-1.628296E-02	3.370945E-03	1.662823E-02	-3.558307E 01	-3.345724E 00	-1.916959E 02
7.542907E 01	-1.185519E-02	2.341769E-03	1.212351E-02	-3.832742E 01	-3.335973E 00	-1.911372E 02
8.684857E 01	-8.740231E-03	1.603039E-03	8.886021E-03	-4.102585E 01	-3.322985E 00	-1.903930E 02
9.999699E 01	-6.455056E-03	1.085393E-03	6.545708E-03	-4.368085E 01	-3.308179E 00	-1.895447E 02

Figure 3-7A FRESP Test One - Tabular Output

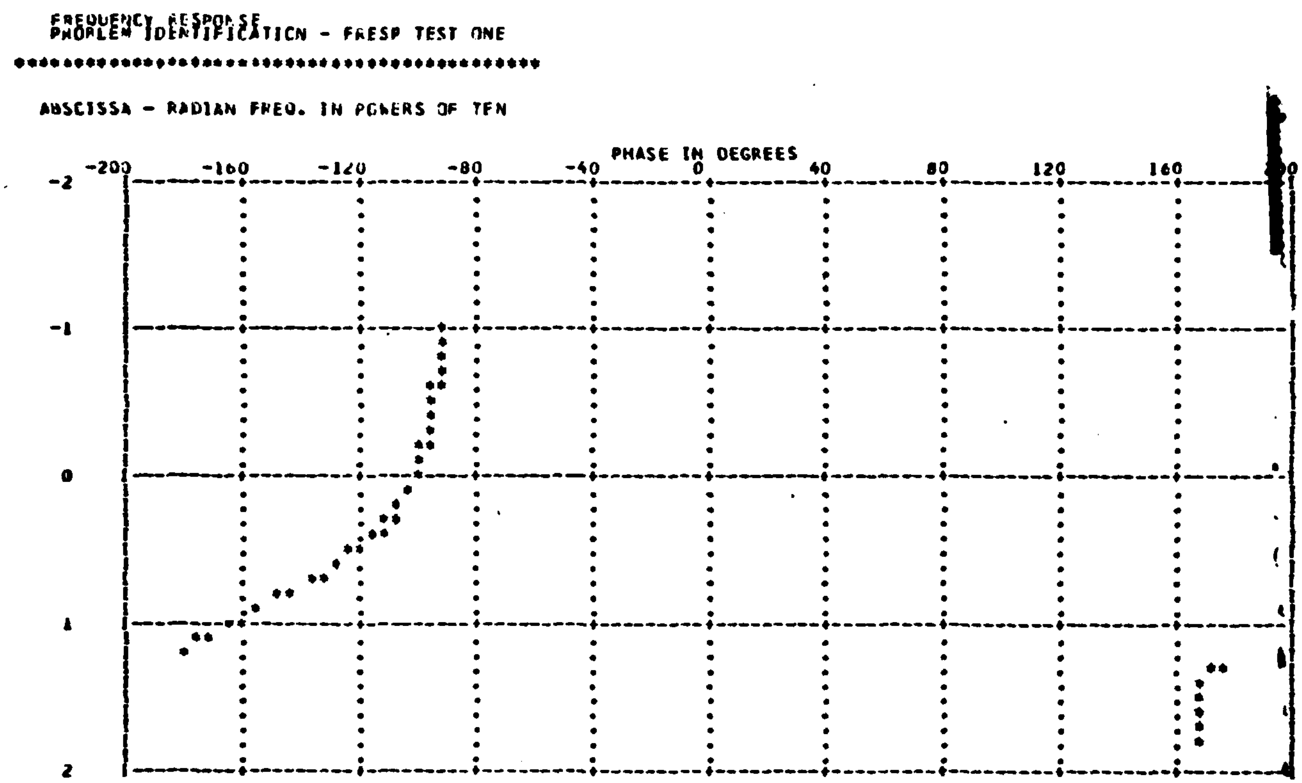


Figure 3-7B FRESP Test One - Bode Plot (Phase)

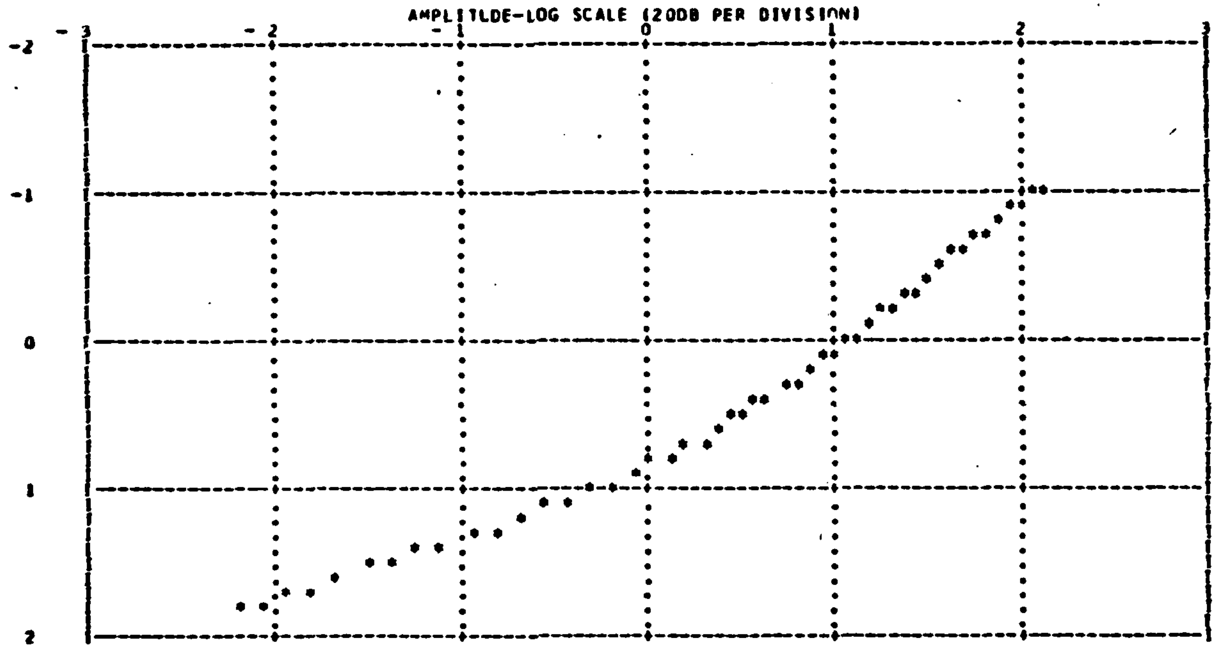


Figure 3-7C FRESP Test One - Bode Plot (Magnitude)

d. Example Two

The problem to be solved now requires that only a Nyquist plot be obtained for the following open-loop transfer function of a compensated system:

$$G(s) = \frac{(s+0.7)(s+0.15)20}{(s+7)(s+0.015)(s+1)(s+2)s}$$

The frequency range is selected to be from 0.2 to 10.0 and calculations are to be carried out for twenty-five frequency values using logarithmic interpolation. The computer deck is then:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(FRESP)
/*
//GO.SYSIN^DD^*
FRESP TEST TWO
20.0
F02
-.15
0.7
F05
0.0
1.0
2.0
0.015
```

```
7.
0.2    10.0    025000001000
/*
```

Results are shown in Figs. 3-8A and 3-8B.

4. Partial Fraction Expansion (PRFEXP)

This subprogram performs the partial fraction expansion of the ratio of two polynomials of the form

$$G(s) = K \frac{N(s)}{D(s)}, \text{ (degree of } N(s) < \text{degree of } D(s))$$

a. Input

The problem identification, the gain value K and the polynomials N(s) and D(s) are entered according to the following input format table:

ENTRY	Input Description	Format	Columns Used
1	Problem identification	5A4	1-20
2	gain value K	8F10.3	1-10
3	letter P or F (for P form and F form), order of N(s) $\leq$ 10	A1, I2	1, 2-3
4	enter N(s) in form specified on previous card	8F10.0	1-10, 10-11, etc.
5	letter P or F (for P form and F form), order of D(s) $\leq$ 10	A1, I2	1, 2-3
6	enter D(s) in form specified on previous card	8F10.0	1-10, 10-11, etc.

Table V - Input Format Table for PRFEXP

```
FREQUENCY RESPONSE
PROBLEM IDENTIFICATION - FRESP TEST TWO
*****************************************************

GAIN =  2.000000E 01

NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S

1.050000E-01 8.500000E-01 1.000000E 00

NUMERATOR ROOTS ARE
   REAL PART           IMAG. PART
-1.500000E-01 0.0
-7.000000E-01 0.0

DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S

0.0           2.099999E-01 1.434500E 01 2.314999E 01 1.001500E 01 1.000000E 00

DENOMINATOR ROOTS ARE
   REAL PART           IMAG. PART
 0.0           0.0
-1.000000E 00 0.0
-2.000000E 00 0.0
-7.000000E 00 0.0
-1.500000E-02 0.0
*****************************************************
```

RADIAN FREQ.	REAL PART	IMAGINARY PART	MAGNITUDE	MAG IN DB	PHASE (RAD)	PHASE (DEG)
2.000000E-01	-3.652469E 00	-5.169011E 00	6.321870E 00	1.601694E 01	-2.186765E 00	-1.252924E 02
2.354076E-01	-2.727261E 00	-4.337989E 00	5.124069E 00	1.413230E 01	-2.132047E 00	-1.221573E 02
2.770837E-01	-2.022156E 00	-3.663237E 00	4.203782E 00	1.247280E 01	-2.083514E 00	-1.193766E 02
3.261381E-01	-1.560852E 00	-3.104766E 00	3.486151E 00	1.084692E 01	-2.043372E 00	-1.170766E 02
3.838769E-01	-1.250196E 00	-2.636493E 00	2.917887E 00	9.301363E 00	-2.013584E 00	-1.153699E 02
4.518376E-01	-1.014977E 00	-2.241604E 00	2.460693E 00	7.821150E 00	-1.995576E 00	-1.143610E 02
5.318393E-01	-8.537239E-01	-1.934016E 00	2.096654E 00	6.389008E 00	-1.992301E 00	-1.141506E 02
6.259845E-01	-7.454925E-01	-1.610935E 00	1.775069E 00	4.984305E 00	-2.004219E 00	-1.148333E 02
7.368077E-01	-6.737026E-01	-1.351984E 00	1.540451E 00	3.587134E 00	-2.033119E 00	-1.164891E 02
8.672506E-01	-6.245273E-01	-1.118862E 00	1.281362E 00	2.153433E 00	-2.079893E 00	-1.191694E 02
1.020737E 00	-5.602252E-01	-9.067413E-01	1.079741E 00	6.663877E-01	-2.144713E 00	-1.228834E 02
1.201505E 00	-5.437801E-01	-7.134911E-01	9.004324E-01	-9.109780E-01	-2.226879E 00	-1.275908E 02
1.414216E 00	-5.070146E-01	-5.348661E-01	7.406235E-01	-2.610048E 00	-2.324828E 00	-1.332025E 02
1.664586E 00	-4.562319E-01	-3.883906E-01	5.991671E-01	-4.449113E 00	-2.436343E 00	-1.395922E 02
1.959281E 00	-3.572385E-01	-2.618510E-01	4.757774E-01	-6.451923E 00	-2.558791E 00	-1.466089E 02
2.306149E 00	-3.331451E-01	-1.614229E-01	3.703676E-01	-8.627340E 00	-2.689414E 00	-1.540571E 02
2.714474E 00	-2.694570E-01	-8.778441E-02	2.824833E-01	-1.098014E 01	-2.825600E 00	-1.618950E 02
3.194951E 00	-2.078038E-01	-3.737412E-02	2.110901E-01	-1.351064E 01	-2.965044E 00	-1.698845E 02
3.760613E 00	-1.544364E-01	-5.530939E-03	1.545853E-01	-1.621663E 01	-3.105804E 00	-1.779495E 02
4.426365E 00	-1.103745E-01	1.159004E-02	1.109818E-01	-1.909496E 01	-3.246214E 00	-1.859544E 02
5.210024E 00	-7.534459E-02	1.881273E-02	7.814294E-02	-2.214220E 01	-3.364727E 00	-1.935306E 02
6.132347E 00	-5.017176E-02	1.993391E-02	5.348672E-02	-2.535425E 01	-3.519780E 00	-2.016686E 02
7.218064E 00	-3.199511E-02	1.781889E-02	3.662241E-02	-2.872505E 01	-3.649736E 00	-2.091145E 02
8.495937E 00	-1.971144E-02	1.441332E-02	2.441894E-02	-3.224544E 01	-3.772963E 00	-2.161749E 02
1.000004E 01	-1.176644E-02	1.088758E-02	1.602747E-02	-3.590268E 01	-3.887985E 00	-2.227652E 02

Figure 3-8A FRESP Test Two - Tabular Output

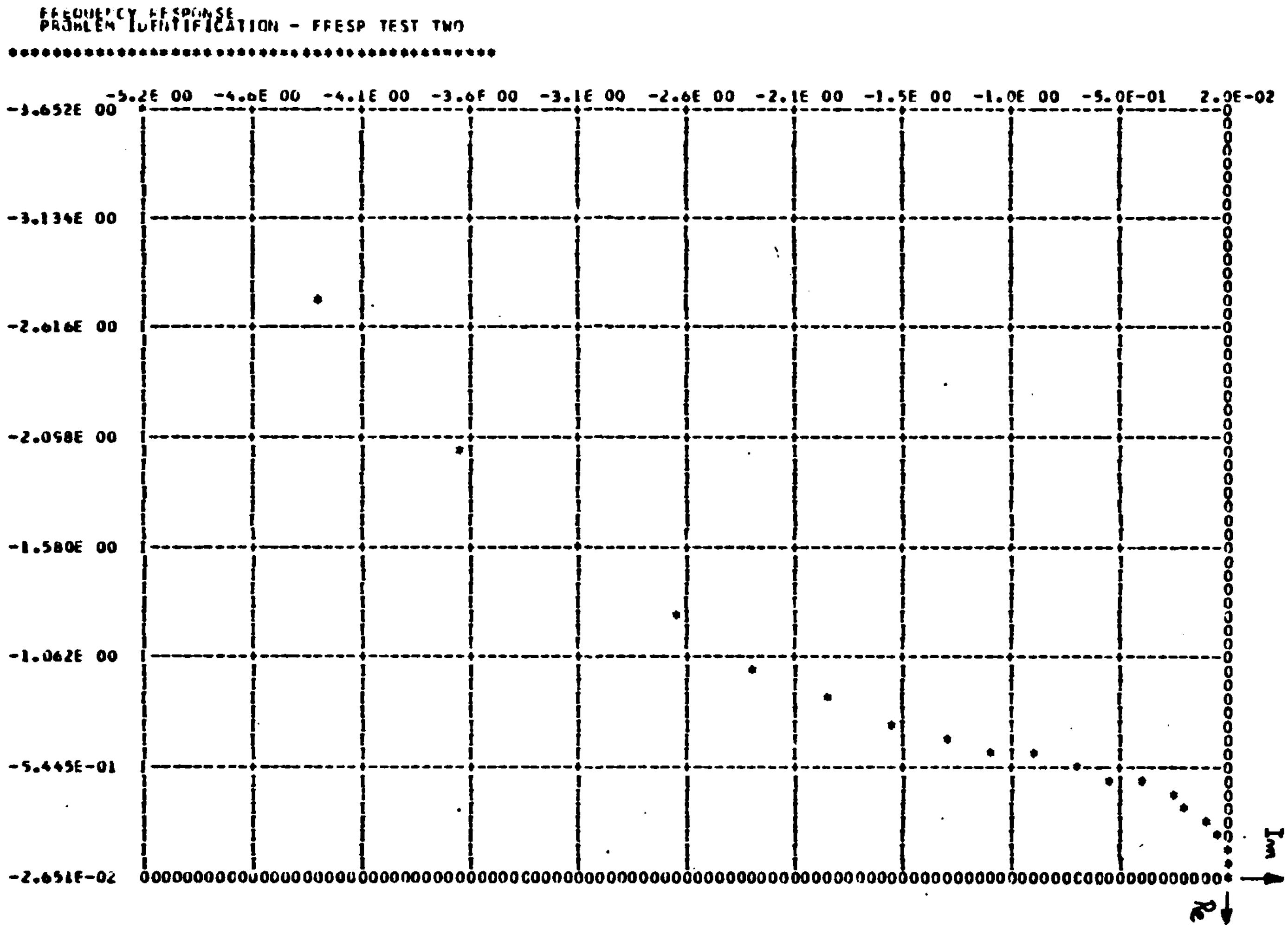


Figure 3-8B FRESP Test Two - Nyquist Diagram

D(s) must not have multiple complex roots for the subprogram to work. If it does, a message is printed and the problem terminates at that point. Note that D(s) may have multiple real roots though.

b. Output

The problem identification and the gain value are listed followed by the numerator and the denominator in both factored and unfactored forms. For the denominator, each root value is listed <u>once only</u> with its multiplicity indicated. Note that roots are considered equal if their real and imaginary parts do not differ by more than 0.005. The example presented in c. illustrates how to deal with multiplicity of roots in the interpretation of the results. The residue matrix real and imaginary parts is then given.

This subprogram can be run as a class A job.

c. Example

The partial fraction expansions of the following rational functions are to be performed:

(a) $$\frac{20}{656 + 752s + 264s^2 + 28s^3 + s^4}$$

(b) $$\frac{2 + s}{2 + 4s + 3s^2 + s^3}$$

N(s) and D(s) are entered using both the F and the P forms and the partial fraction expansions of the two polynomial ratios can easily be obtained in a single run by stacking the data deck.

The computer cards are:

```
//  (standard OS JOB card)
// EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(PRFEXP)
/*
//GO.SYSIN^DD^*
PARTIAL FRACTION A
20.
P00
1.0
P04
656.     752.     264.     28.     1.0
PARTIAL FRACTION B
1.0
F01
2.0
P03
2.0      4.0      3.0      1.0
/*
```

and the solutions are presented in Figs. 3-9A and 3-9B. Interpretation of these results gives:

(1) Partial fraction A $= \dfrac{.0139 + j.0124}{s+12+j4.47} + \dfrac{.0139 - j.0124}{s+12-j4.47}$

$$+ \frac{-.0278}{s+2} + \frac{.1667}{(s+2)^2}$$

```
PARTIAL FRACTION EXPANSION
PROBLEM IDENTIFICATION -   PARTIAL FRACTION A

NUMERATOR GAIN =  2.000E 01

NUMERATOR COEFF. - IN ASCENDING POWERS

 1.000E 00

DENOMINATOR COEFF. - IN ASCENDING POWERS

 6.560E 02 7.520E 02 2.640E 02 2.800E 01 1.000E 00

*********************************************

DENOMINATOR ROOTS

  REAL PART       IMAG. PART   MULTIPLICITY
-1.1999992E 01-4.4721594E 00      1
-1.1999992E 01 4.4721594E 00      1
-2.0011177E 00 0.0                2

*********************************************

RESIDUE MATRIX - REAL PART

 1.3892505E-02
 1.3892505E-02
-2.7785011E-02 1.6669762E-01

RESIDUE MATRIX - IMAG. PART

 1.2423638E-02
-1.2423638E-02
 0.0           0.0

*********************************************
```

Figure 3-9A Partial Fraction Expansion A

```
PARTIAL FRACTION EXPANSION
PROBLEM IDENTIFICATION -   PARTIAL FRACTION B

NUMERATOR GAIN =  1.000E 00

NUMERATOR COEFF. - IN ASCENDING POWERS

 2.000E 00 1.000E 00

NUMERATOR ROOTS

  REAL PART           IMAG. PART
-2.0000000E 00 0.0

DENOMINATOR COEFF. - IN ASCENDING POWERS

 2.000E 00 4.000E 00 3.000E 00 1.000E 00

*********************************************

DENOMINATOR ROOTS

  REAL PART       IMAG. PART   MULTIPLICITY
-1.0000000E 00-1.0000010E 00      1
-1.0000000E 00 1.0000010E 00      1
-9.9999994E-01 0.0                1

*********************************************

RESIDUE MATRIX - REAL PART

-4.9999905E-01
-4.9999905E-01
 9.9999803E-01

RESIDUE MATRIX - IMAG. PART

 4.9999928E-01
-4.9999928E-01
 0.0

*********************************************
```

Figure 3-9B Partial Fraction Expansion B

Note that the second residue appearing in the output belongs to $(s+2)^2$. If a multiplicity three had been the case, a third residue would have been the numerator of a cubic.

(2) Partial fraction B = $\frac{-.5+j.5}{s+1+j} + \frac{-.5-j.5}{s+1-j} + \frac{1}{s+1}$

5. Roots of a Polynomial (Roots)

This subprogram finds the roots of a polynomial of degree less or equal to twenty.

a. Input

The first data card contains the problem identification in the first twenty columns and the polynomial order in columns 21-22 (format I2). On the next card(s) the polynomial coefficients starting with the lowest order term are entered (format 8E10.0). These two entries are repeated for every polynomial to be factored. Note that the highest order term coefficient must be unity.

Entry	Input Description	Format	Columns Used
1	Problem identification, polynomial order	5A4 I2	1-20 21-22
2	polynomial coefficients in ascending order (highest order term coefficient being one)	8E10.0	1-10, 11-20, 21-30, etc.

b. Output

The problem identification and the polynomial coefficients are listed for reference. The roots' real and imaginary parts are then printed.

c. Example

The following polynomials are to be factored:

$$s^3 + 1$$

$$s^4 + s^3 + 12s^2 - 5s + 1$$

$$s^5 + s^2 + s$$

The computer cards are then:

```
//(standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^ ^  INCLUDE^SYSLIB(ROOTS)
/*
//GO.SYSIN^DD^*
Roots test one    03
1.0    0.0    0.0    1.0
Roots test two    04
1.0    -5.0   12.0   1.0    1.0
Roots test three
0.0    1.0    1.0    0.0    0.0    1.0
/*
```

The result is shown in Figure 3-10.

C. TIME RESPONSE AND MATRIX MANIPULATION SUBPROGRAMS

These three subprograms permit a user to analyze linear control systems for rational and graphical time response and

```
ROOTS OF A POLYNOMIAL
PROBLEM IDENTIFICATION -      ROOTS TEST ONE
POLYNOMIAL COEFFICIENTS - IN ASCENDING POWERS OF S
 1.0000000E 00      0.0                0.0                 1.0000000E 00
THE ROOTS ARE                 REAL PART          IMAGINARY PART
                           -1.0000000E 00        0.0
                            5.0000000E-01        8.6602539E-01
                            5.0000000E-01       -8.6602539E-01
**************************************************

ROOTS OF A POLYNOMIAL
PROBLEM IDENTIFICATION -      ROOTS TEST TWO
POLYNOMIAL COEFFICIENTS - IN ASCENDING POWERS OF S
 1.0000000E 00     -5.0000000E 00      1.2000000E 01       1.0000000E 00       1.0000000E 00
THE ROOTS ARE                 REAL PART          IMAGINARY PART
                           -7.0456743E-01        3.4641142E 00
                           -7.0456743E-01       -3.4641142E 00
                            2.0456773E-01        1.9538277E-01
                            2.0456773E-01       -1.9538277E-01
**************************************************

ROOTS OF A POLYNOMIAL
PROBLEM IDENTIFICATION -      ROOTS TEST THREE
POLYNOMIAL COEFFICIENTS - IN ASCENDING POWERS OF S
 0.0               1.0000000E 00       1.0000000E 00       0.0                 0.0                 1.0000000E 00
THE ROOTS ARE                 REAL PART          IMAGINARY PART
                           -7.2713614E-01        4.3001431E-01
                           -7.2713614E-01       -4.3001431E-01
                            7.2713614E-01        9.3409908E-01
                            7.2713614E-01       -9.3409908E-01
                            0.0                  0.0
**************************************************
```

Figure 3-10 Roots of a Polynomial - Three Tests

also provide matrix manipulation to easily solve for determinants, inverses, state transition and resolvent matrices, eigenvalues and characteristic polynomials.

The control system must be linear and represented in state variable form as [1]

$$\dot{\underset{\sim}{x}}(t)=\underset{\sim}{A}\ \underset{\sim}{x}(t) + \underset{\sim}{b}\ u(t)$$

$$u(t)=K[r(t) - \underset{\sim}{k}^T\underset{\sim}{x}(t)]$$

$$y(t)=\underset{\sim}{c}\ \underset{\sim}{x}(t)$$

where u(t), r(t) and K are scalar and the system order is less or equal to ten. In block diagram form, the matrix system can be represented as

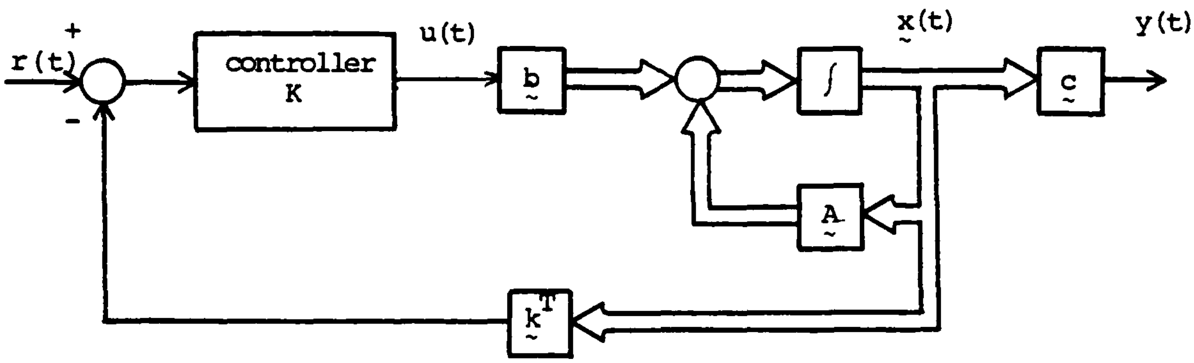


Fig 3-11 Linear Control System Block Diagram.

The diagram readily shows all the elements to be provided for the study of any given system. For instance one can view that setting $\underset{\sim}{k}^T = \underset{\sim}{0}$ gives an open-loop system and that unforced system analysis can be done by simply letting r(t) = 0.

1. <u>Basic Matrix Manipulation (BASMAT)</u>

This subprogram is used to perform various calculations associated with the plant matrix $\underset{\sim}{A}$ of a given linear control system. It is a class A job and must be run under Mode One or Mode Three.

a. Input

The problem identification and the dimension of $\underset{\sim}{A}$ are given on the first card. Next the $\underset{\sim}{A}$ matrix is entered, one row at a time using an 8E10.5 format. Thus, if the dimension of the matrix is eight or less, one row per card. Otherwise the 9th and/or 10th elements appear on a second card and the rule becomes one row per two cards. The last card indicates what matrix operations are to be performed. The key to obtain the proper results is explained after the input format table.

Entry	Input Description		Format	Columns Used
1	Problem identification dimension of $\underset{\sim}{A}(N \leq 10)$		5A4,I2	1, 2-3
2	$\underset{\sim}{A}(N \times N)$ matrix (one row per card		8E10.5	1-10,11-20,etc.
	for $N \leq 8$; one row per two cards for $N > 8$)		8E10.5	1-10,11-20,etc.
3	option det	0, determinant desired	I1	1,
		1, determinant not desired	I1	
	option inv	0, inverse desired	I1	2,
		1, inverse not desired		

Entry	Input Description		Format	Columns Used
	option phi(s)[1]	0, $\underset{\sim}{\phi}(s)$ desired	I1	3,
		1, $\underset{\sim}{\phi}(s)$ not desired		
	option C.E.	0, characteristic polynomial desired	I1	4,
		1, characteristic polynomial not desired		
	option eigen	0, eigenvalues desired	I1	5,
		1, eigenvalues not desired		
	option phi(t)	0, $\underset{\sim}{\phi}(t)$ desired	I1	6.
		1, $\underset{\sim}{\phi}(t)$ not desired		

Table VI - Input Format Table for BASMAT

Thus, a zero indicates that the computation is desired while a number from 1 to 9 informs that the listed operation is not to be performed. Six zeros or a blank card would result in an output that contains the $\underset{\sim}{A}$ matrix determinant, inverse, resolvent, characteristic polynomial, eigenvalues and state transition matrix.

b. Output

The problem identification and the $\underset{\sim}{A}$ matrix are listed first. Then the result of each operation selected on the option card is printed as follows:

(1) det $(\underset{\sim}{A})$ - a scalar

(2) $\underset{\sim}{A}^{-1}$ - a matrix presented one row at a time

(3),(4) resolvent matrix and characteristic polynomial.

---

[1] $\underset{\sim}{\phi}(s) \triangleq [s\underset{\sim}{I} - \underset{\sim}{A}]^{-1}$, which is called the resolvent matrix, is the Laplace transform of the state transition matrix $\underset{\sim}{\phi}(t) = e^{\underset{\sim}{A}t}$.

The coefficient matrix of the numerator of the resolvent matrix appears first, followed by the characteristic polynomial in ascending powers of s.

(5) Eigenvalues - listed indicating the real and imaginary parts

(6) Time domain state transition matrix - $\underset{\sim}{\phi}(t)$ (see part c, example two).

The subprogram is restricted by the fact that $\underset{\sim}{\phi}(t)$ cannot be calculated if eigenvalues are multiple. If a situation where the state transition matrix is requested where eigenvalues are not simple, a message is printed (see part c, example one) and the computer goes to the next problem. Note that eigenvalues are considered to be identical if their real parts and their imaginary parts differ by less than 0.005.

c. Examples

(1) Example One

The resolvent matrix $(s\underset{\sim}{I} - \underset{\sim}{A})^{-1}$ and the state transition matrix $\underset{\sim}{\phi}(t)$ are to be found for the plant matrix

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -2 \end{bmatrix}$$

Thus, the matrix has dimension N = 3. The options are set equal to the following values:

option det = 1	determinant value is not desired
option inv = 1	inverse $\underset{\sim}{A}^{-1}$ is not to be calculated
option phi(s) = 0	$\underset{\sim}{\phi}(s)$ is desired
option C.E. = 0	characteristic polynomial is desired
option eigen = 0	eigenvalues are to be computed
option phi(t) = 0	$\underset{\sim}{\phi}(t)$ is desired.

The computer card deck is then:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(BASMAT)
/*
//GO.SYSIN^DD^*
BASMAT TEST ONE 03
0.0     1.0     0.0
0.0     0.0     1.0
0.0     0.0     -2.0
110000
/*
```

The computer results shown in Fig 3-12 can be interpreted as follows:

$$\underset{\sim}{\phi}(s) = \frac{1}{(s^3+2s^2)}\begin{bmatrix} s^2+2s & s+2 & 1 \\ 0 & s^2+2s & s \\ 0 & 0 & s^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} & \frac{1}{s^2(s+2)} \\ 0 & \frac{1}{s} & \frac{1}{s(s+2)} \\ 0 & 0 & \frac{1}{s+2} \end{bmatrix}$$

```
BASIC MATRIX PROGRAM
PROBLEM IDENTIFICATION-     BASMAT TEST ONE
*******************************************

     THE A MATRIX

      4.0000000E 00        2.0000000E 00        1.0000000E 00
      0.0                  6.0000000E 00        1.0000000E 00
      0.0                 -4.0000000E 00        2.0000000E 00
*******************************************

     THE MATRIX COEFFICIENTS OF THE NUMERATOR OF THE RESOLVENT MATRIX

     THE MATRIX COEFFICIENT OF S**2

      1.0000000E 00        0.0                  0.0
      0.0                  1.0000000E 00        0.0
      0.0                  0.0                  1.0000000E 00

     THE MATRIX COEFFICIENT OF S**1

     -8.0000000E 00        2.0000000E 00        1.0000000E 00
      0.0                 -6.0000000E 00        1.0000000E 00
      0.0                 -4.0000000E 00       -1.0000000E 01

     THE MATRIX COEFFICIENT OF S**0

      1.5999985E 01       -3.0000000E 00       -4.0000000E 00
      0.0                  7.9999847E 00       -4.0000000E 00
      0.0                  1.6000000E 01        2.3999985E 01
*******************************************

     THE CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S

     -6.3999985E 01       4.7999985E 01       -1.2000000E 01        1.0000000E 00
*******************************************

     THE EIGENVALUES OF THE A MATRIX
        REAL PART            IMAGINARY PART

      4.0001345E 00        0.0
      4.0001354E 00        0.0
      3.9997292E 00        0.0

     ***WARNING***

    EIGENVALUES MUST BE SIMPLE
    CALCULATIONS CANNOT BE COMPLETED FOR THIS PROBLEM
***************************************
```

Figure 3-12 BASMAT Test One

The state transition matrix $\underset{\sim}{\phi}(t)$ cannot be obtained since the eigenvalues are not simple.

(2) Example Two

This second example shows the complete solution, i.e., determinant, inverse, resolvent matrix, characteristic polynomial, eigenvalues and state transition matrix, for a case where

$$\underset{\sim}{A} = \begin{bmatrix} 2.0 & 2.2 & 2.5 \\ 5.1 & 3.4 & 7.1 \\ 0.9 & 1.1 & 1.1 \end{bmatrix}$$

Since all the calculations are requested, the option card is left blank. The card deck is

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(BASMAT)
/*
//GO.SYSIN^DD^*
BASMAT TEST TWO 03
2.0     2.2     2.5
5.1     3.4     7.1
0.9     1.1     1.1
(blank card)
/*
```

Results appear in Fig 3-13. Interpretation of these results is fairly straightforward. For instance, the first term of the resolvent matrix, $\underset{\sim}{\phi}(s)$, is

$$\phi_{11}(s) = \frac{s^2 - 4.5s - 4.07}{s^3 - 6.5s^2 - 8.54\,s + 0.049}$$

and, similarly, the first term of the transition matrix, $\underset{\sim}{\phi}(t)$, is

$$\phi_{11}(t) = (0.475e^{+0.0057t} + 0.229e^{-1.13t} + 0.296e^{+7.62t})$$

2. Rational Time Response (RTRESP)

This subprogram may be used whenever it is desired to obtain the time response in closed form [1] of a linear control system described by the following set of equations:

$$\dot{x}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$u(t) = K[r(t) - \underset{\sim}{k}^T\,x(t)]$$

$$y(t) = \underset{\sim}{c}\,\underset{\sim}{x}(t)$$

The system can have any initial conditions $\underset{\sim}{x}(t_o)$ but the scalar forcing function $r(t)$ must have a rational Laplace transform such that

$$[r(t)] = R(s) = G\,\frac{N(s)}{D(s)}\text{ , where G is a constant,}$$

```
BASIC MATRIX PROGRAM
PROBLEM IDENTIFICATION-      BASMAT TEST TWO

********************************************

     THE A MATRIX

      2.0000000E 00       2.1999998E 00       2.5000000E 00
      5.3999994E 00       3.3999996E 00       7.0999994E 00
      8.9999998E-01       1.0999994E 00       1.0999994E 00

     THE DETERMINANT OF THE MATRIX

     -4.9001485E-02

     THE INVERSE OF THE MATRIX

      6.3061829E 01      -6.7347212E 00      -1.4530740E 02
     -1.5916466E 01       1.0204105E 00       2.9592041E 01
     -5.2041224E 01       4.4898157E 00       9.0204926E 01

********************************************

     THE MATRIX COEFFICIENTS OF THE NUMERATOR OF THE RESOLVENT MATRIX

     THE MATRIX COEFFICIENT OF S**2

      1.0000000E 00      -3.0517578E-05      -1.5258789E-05
      0.0                 1.0000000E 00      -1.5258789E-05
      0.0                 6.1035156E-05       1.0000000E 00

     THE MATRIX COEFFICIENT OF S**1

     -4.4999990E 00       2.1999998E 00       2.5000000E 00
      5.3999994E 00      -3.0999994E 00       7.0999994E 00
      8.9999998E-01       1.0999994E 00      -5.3999996E 00

     THE MATRIX COEFFICIENT OF S**0

     -4.0699921E 00       3.2999992E-01       7.1199961E 00
      7.7997733E-01      -4.9944669E-02      -1.4500122E 00
      2.5499964E 00      -2.1999931E-01      -4.4199924E 00

********************************************

     THE CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S

      4.9001485E-02      -8.5399904E 00      -6.4999990E 00       1.0000000E 00

********************************************

     THE EIGENVALUES OF THE A MATRIX
        REAL PART           IMAGINARY PART

      5.7130642E-03       0.0
     -1.1256170E 00       0.0
      7.6199007E 00       0.0

********************************************

     THE ELEMENTS OF THE STATE TRANSITION MATRIX

     THE MATRIX COEFFICIENT OF EXP( 5.713064E-03)T

      4.7545737E-01      -3.9706232E-02      -9.2820415E-01
     -9.3932746E-02       7.8541597E-03       1.6361809E-01
     -2.9662697E-01       2.4509659E-02       5.1663453E-01

     THE MATRIX COEFFICIENT OF EXP(-1.125617E 00)T

      2.2865254E-01      -2.1693349E-01       4.3520546E-01
     -5.0137544E-01       4.7568274E-01      -9.5429707E-01
      1.5533940E-01      -1.4737860E-01       2.9566765E-01

     THE MATRIX COEFFICIENT OF EXP( 7.619901E 00)T

      2.9589075E-01       2.5670183E-01       3.9299810E-01
      5.9530617E-01       5.1646215E-01       7.9067850E-01
      1.4128095E-01       1.2256514E-01       1.8764746E-01
```

Figure 3-13 BASMAT Test Two

and

$$N(s) = a_0 + a_1 s + a_2 s^2 + \ldots + s^{\ell}$$

$$D(s) = b_0 + b_1 s + b_2 s^2 + \ldots + s^m$$

with $m > \ell \geq 0$.
Arrange the polynomials so the coefficients in the highest order terms of both N(s) and D(s) are unity and select the input gain G as required.

In addition to the above, it is necessary that the total order of the system, i.e. order of D(s) plus dimension of $\tilde{A}$ be smaller than or equal to ten. This limitation is not overly restrictive but must be taken into account when handling large order systems.

a. Input

The system matrices, feedback coefficients and the controller gain are entered immediately after the problem identification and system order card. The $\tilde{A}$ matrix elements are presented one row at a time. The transpose control vector $\tilde{b}^T$, the output vector $\tilde{c}$, the feedback coefficients k_1, k_2, k_3, ..., k_n and the controller gain K are given using an 8F10.4 format.

Next the initial conditions $x_1(0)$, $x_2(0)$, ..., $x_n(0)$, the input gain G and the numerator and the denominator input polynomials are entered. Both N(s) and D(s) may be entered in factored (F) form or unfactored (P) form and it is

noted that the degree of D(s) must be strictly larger than the degree of N(s).

It is suggested that a signal flow graph, or at least a matrix block diagram, be sketched before an attempt is made to run this subprogram. It does not take long to do so and much can be gained.

The execution time for the subprogram is less than 20 seconds for most cases (class A).

Entry	Input Description	Format	Columns Used
1	Problem identification, order of the system ($N \leq 10$)	5A4, I2	1-20, 21-22
2	plant matrix $\underset{\sim}{A}$ (one row per card if $N \leq 8$; one row per two cards if $N > 8$)	8F10.4	1-10, 11-20, etc.
3	Control matrix $\underset{\sim}{b}^T (1 \times N)$ (on one card if $N \leq 8$; two cards if $N > 8$)	8F10.4	1-10, 11-20, etc.
4	Output vector $\underset{\sim}{c} (1 \times N)$ (on one card if $N \leq 8$, on two cards if $N > 8$)	8F10.4	1-10, 11-20, etc.
5	feedback coefficients $k_1, k_2, \ldots, k_n$ (on one card if $N \leq 8$; on two cards if $N > 8$)	8F10.4	1-10, 11-20, etc.
6	Controller gain K	8F10.4	1-10
7	Initial condition $x_1(0), x_2(0), \ldots, x_n(0)$ (on one card if $N \leq 8$, on two cards if $N > 8$)	8F10.4	1-10, 11-20, etc.
8	Input gain G	8F10.4	1-10
9	Letter P or F (for P form or F form), polynomial order $\ell \leq M$	A1, I2	1, 2-3

Entry	Input Description	Format	Columns Used
10	Enter N(s) in format specified on the previous card.	8F10.4	1-10, 11-20, etc.
11	Letter P or F (for P form or F form), polynomial order M $\leq$ 10	A1, I2	1, 2-3
12	Enter D(s) in format specified on the previous card.	8F10.4	1-10, 11-20, etc.

Table VII - Input Format Table for RTRESP

b. Output Format

All the information given as input is repeated for reference. The polynomials N(s) and D(s) are presented both in factored and unfactored forms.

The rational time response of each component of the state vector $\underset{\sim}{x}(t)$ and the scalar output y(t) are printed in pseudo-matrix form. Here again a hypothetical example can clarify the presentation. For a two-state problem, assuming complex poles and a step input, the computer output would look like:

THE TIME RESPONSE OF THE STATE X(t)

THE VECTOR COEFFICIENT OF EXP(A)T * COS(B)T

x_{11} $\qquad$ x_{12}

THE VECTOR COEFFICIENT OF EXP(A)T * SIN(B)T

x_{21} $\qquad$ x_{22}

THE VECTOR COEFFICIENT OF EXP(0.0)T

x_{31} $\qquad$ x_{32}

where x_{11}, x_{12}, x_{21}, x_{22}, x_{31} and x_{32} are numbers. The result would be interpreted as:

$$x_1(t) = x_{11} * \exp(at) * \cos(bt) + x_{21} * \exp(at) * \sin(bt) + x_{31}$$

$$x_2(t) = x_{12} * \exp(at) * \cos(bt) + x_{22} * \exp(at) * \sin(bt) + x_{32}$$

The procedure to obtain y(t) is the same. Note that if more than one output y(t) is desired, the subprogram must be rerun changing the $\underset{\sim}{c}$ matrix each time.

c. Example

The open-loop rational time response is desired for

$$\frac{Y(s)}{X(s)} = \frac{.1923}{s^2 + 2.346s + 3.846}$$

The first step is to get the signal flow graph and state equations.

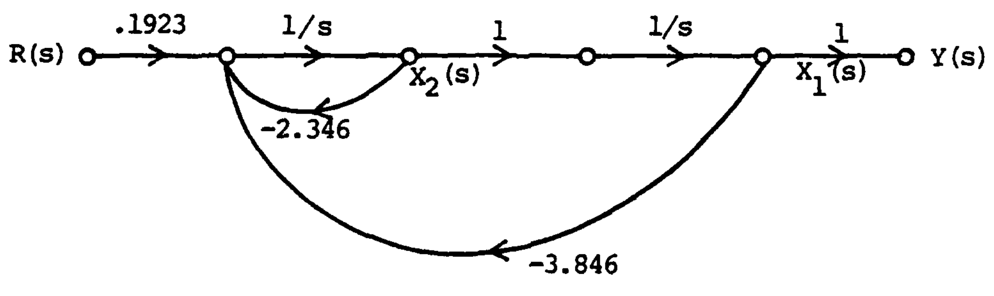


Fig 3-14 Control System for RTRESP Test

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -3.846x_1(t) - 2.346x_2(t) + .1923u(t)$$

$$u(t) = r(t)$$

$$y(t) = x_1(t)$$

The data from the system is then:

$$\underset{\sim}{A} = \begin{bmatrix} 0.0 & 1.0 \\ -3.846 & -2.346 \end{bmatrix}$$

$$\underset{\sim}{b}^T = [0.0 \quad .1923]$$

$$\underset{\sim}{c} = [1.0 \quad 0.0]$$

$$\underset{\sim}{k}^T = [0.0 \quad 0.0]$$

$$K = 1.0$$

$$\underset{\sim}{x}(0) = \underset{\sim}{0}$$

The system time response in closed form is required for a step input of magnitude 2. Thus

$$R(s) = \frac{2}{s}$$

and the data are

input gain = 2

N(s) = 1

D(s) = s

The control and data cards to run the program are as follow:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD *
^^INCLUDE^SYSLIB(RTRESP)
/*
//GO.SYSIN^DD^*
RTRESP TEST          02
0.0       1.0
-3.836    -2.346
0.0       0.1923
1.0       0.0
0.0       0.0
1.0
0.0       0.0
2.0
P00
1.0
F01
0.0
/*
```

The results shown in Fig. 3-15 are interpreted as:

$$x_1(t) \simeq -.1 * \exp(-1.173t) * \cos(1.57t) - .075 * \exp(-1.173t) * \sin(1.57t) + 0.1$$

$$x_2(t) \simeq 0.245 * \exp(-1.173t) * \sin(1.57t)$$

$$y(t) = x_1(t)$$

3. Graphical Time Response (GTRESP)

The subprogram is a slightly modified version of the one presented by Melsa and Jones [1]. It still determines the time response of the closed loop system

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$u(t) = K[r(t) - \underset{\sim}{k}^T\,\underset{\sim}{x}(t)]$$

$$y(t) = \underset{\sim}{c}\,\underset{\sim}{x}(t)$$

with initial conditions $\underset{\sim}{x}(t_0)$ and displays the results both in tabular and graphical forms. However, instead of having all the desired plots drawn on one graph only, it also produces one graph for every selected variable.

The subprogram solves linear systems. It is a Class B job when graphical output is requested but reduces to a Class A job when tabular output only is to be listed. The subprogram must be accessed under Mode Two and requires an exterior subroutine to define the scalar forcing input r(t).

```
RATIONAL TIME RESPONSE
PROBLEM IDENTIFICATION - RTRESP TEST
*********************************************

   THE A MATRIX

         0.0                  1.0000000
        -3.8459997           -2.3459997

   THE B MATRIX

         0.0                  0.1923000

   THE C MATRIX

         1.0000000            0.0

   FEEDBACK COEFF.

         0.0                  0.0

   GAIN =  1.0000000E 00
*********************************************

   INITIAL CONDITIONS - X(0)

         0.0                  0.0

   RGAIN =  2.0000000E 00

   NUMERATOR POLYNOMIAL OF R(S) - ASCENDING POWERS OF S
         1.0000000

   DENOMINATOR POLYNOMIAL OF R(S) - ASCENDING POWERS OF S
         0.0                  1.0000000

   DENOMINATOR ROOTS ARE
     REAL PART          IMAG. PART
         0.0                  0.0
*********************************************

   THE TIME RESPONSE OF THE STATE X(T)

   THE VECTOR COEFFICIENT OF EXP(-1.172999E 00)T*COS(1.571647E 00)T
        -0.1000000            0.0000001

   THE VECTOR COEFFICIENT OF EXP(-1.172999E 00)T*SIN(1.571647E 00)T
        -0.0746350            0.2447114

   THE VECTOR COEFFICIENT OF EXP( 0.0          )T
         0.0599995           -0.0000001
*********************************************

   THE TIME RESPONSE OF THE OUTPUT Y(T)

   THE COEFFICIENT OF EXP(-1.172999E 00)T*COS(1.571647E 00)T
        -0.1000000

   THE COEFFICIENT OF EXP(-1.172999E 00)T*SIN(1.571647E 00)T
        -0.0746350

   THE COEFFICIENT OF EXP( 0.0          )T
         0.0599995
*********************************************
```

Figure 3-15 RTRESP Test - Computer Output

a. Input

The first element to be input is the forcing function r(t). A short defining subroutine must be written in the following manner:

```
SUBROUTINE RFIND(T,R)
(FORTRAN statements defining r(t))
(Example:   R = 2.5*T+SIN(4.2*T)
RETURN
END
```

Next the remaining parameters are entered as a data deck which closely resembles the one for RTRESP. The problem identification and system order ($N \leq 10$) are given on the first card. Then the $N \times N$ plant matrix $\underset{\sim}{A}$, the single row matrix $\underset{\sim}{b}^T$, the output matrix $\underset{\sim}{c}$, the feedback coefficient matrix $\underset{\sim}{k}^T$, the controller gain K and the initial conditions $\underset{\sim}{x}(t_0)$ are presented as indicated on the input format table. The next-to-last card specifies the time factors: the initial time, the final time, the integration step size and the frequency of output are given in an 8E10.0 format. The last card enumerates the variables to be plotted versus time.

Here some specifics regarding the time specifications and the variables to be plotted must be remembered.

(1) Common sense must be used when selecting the initial and final time. Intelligent guesses should be made based on experience and the system dynamics.

(2) The integration step size is also related to the system dynamics. It should be small enough to give a precise solution but not excessively small as to increase the computing time unnecessarily. As a rule of thumb one can start by letting the integration step size be

$$DT = \frac{(\text{final time}) - (\text{initial time})}{1000.}$$

(3) The frequency of output (FREQ) determines both the number of points to be plotted in the total time interval and the physical dimension of the graph. The formula to determine the value of FREQ is

$$FREQ = \frac{(\text{final time}) - (\text{initial time})}{(\text{integration step size})(\text{number of points to be plotted}}$$

$$= (\text{No. of time steps})/(\text{No. of points plotted})$$

where the number of points to be plotted must <u>always</u> be less than or equal to 100. Equivalently one can say that the plotting is constrained by the equation

$$FREQ \geq \frac{(\text{final time}) - (\text{initial time})}{(\text{integration step size})(100)}$$

This relationship is very important. It restricts the user but also permits him to establish in advance the number of points to be plotted per curve and the scaling of the time axis. This is illustrated by the following example.

Assume that the initial time is 0.0, the final time is 10.0 and the step size is 0.005. What value should be

used for the frequency of output, FREQ? Using the rule stated above,

$$\text{FREQ} \geq \frac{(10.0 - 0.0)}{(.005)(100)} = 20$$

Thus the frequency of output must be greater than or equal to twenty. Expecting a moderately oscillating time response, a "number of points to be plotted" equal to fifty is decided upon. Thus,

$$\text{FREQ} = \frac{(10 - 0)}{(.005)(50)} = 40$$

giving a sampling interval (S.I.)

$$\text{S.I.} = (\text{FREQ})(\text{Step Size}) = (40)(.005) = 0.2$$

In summary, for this example, setting FREQ equal to 40 would give an output of 50 points, each 0.2 seconds apart between the initial value TI = 0.0 and the final value TF = 10.0 seconds.

Note that the physical dimension of the graph is *directly proportional* to the number of points to be plotted. Fifty points usually gives a good drawing and is suggested as starting value.

(4) Approximate equations for the graph dimension are presented as extra information only. These do not help to solve the problem but give an idea of what to expect:

dependent variable or y axis = 36 cm (fixed)

independent variable or t axis = (.318)×(number of points) cm

The last card of the data deck indicates what dependent variables are to be plotted. A maximum of eight graphs can be output for every program run. If tabular output only are desired, the last card is left blank. The variables for which time responses are to be drawn are specified by giving the symbol that corresponds to the desired variable:

Symbol	Variable to be plotted	Symbol	Variable to be plotted
1	x1(t)	8	x8(t)
2	x2(t)	9	x9(t)
3	x3(t)	S	x10(t)
4	x4(t)	R	error signal
5	x5(t)	U	controller input
6	x6(t)	Y	output
7	x7(t)	R	forcing input

Table VIII - Symbol Indicating Variables to be Plotted by GTRESP

where the error signal is defined as

$$e(t) = r(t) - y(t)$$

All the above is summarized by the following table:

Entry	Input Description	Format	Columns Used
1	Problem identification, order of the system ($N \leq 10$)	5A4, I2	1-20, 21-22
2	Plant matrix $A(N \times N)$ (one row per card if $N \leq 8$ or one row per two cards if $N > 8$)	8E10.0	1-10, 11-20, etc.
3	Distribution matrix b^T $(1 \times N)$ (one card if $N \leq 8$ or two cards if $N > 8$)	8E10.0	1-10, 11-20, etc.
4	Output vector c $(1 \times N)$ (one card if $N \leq 8$ or two cards if $N > 8$)	8E10.0	1-10, 11-20, etc.
5	feedback coefficients k_1, k_2, ..., k_n (one card if $N \leq 8$ or two cards if $N > 8$)	8E10.0	1-10, 11-20, etc.
6	Controller gain K	8E10.0	1-10
7	Initial condition $x_1(t_o)$, $x_2(t_o)$, ..., $x_n(t_o)$ (on one card if $N \leq 8$ or two cards if $N > 8$)	8E10.0	1-10, 11-20, etc.
8	Initial time TI, final time TF, step size DT, frequency of output FREQ	8E10.0	1-10, 11-20, 21-30, 31-40.
9	Any of the following symbols in any of the first eight columns of the card (maximum of 8): Y,R,U,E,1,2,3,4,5,6,7,8,9,A	8A1	1,2,3,4,5,6,7,8

Table IX - Input Format Table for GTRESP

b. Output

The problem identification, $\underset{\sim}{A}$, $\underset{\sim}{b}^T$, $\underset{\sim}{c}$, $\underset{\sim}{k}^T$, K, $\underset{\sim}{x}(t_o)$, the initial time TI, the final time TF, the integration step size DT and the frequency of output FREQ are printed out for future reference. Then the tabular output of all the state

variables together with the control input u(t) and the output y(t) are listed versus time. Finally, the graphical outputs are given. As mentioned earlier, one graph is produced for each selected variable. At the end of the run, a compact solution is presented by plotting all the curves on a single graph.

c. Example

An uncompensated system is described by

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = u(t)$$

The system is compensated by feeding back both states and the graphical time response is to be obtained for initial condition only. The initial conditions are $x_1(0) = 10.0$ and $x_2(0) = 0.0$. The controller gain equals 1.6.

The following diagram represents the complete system:

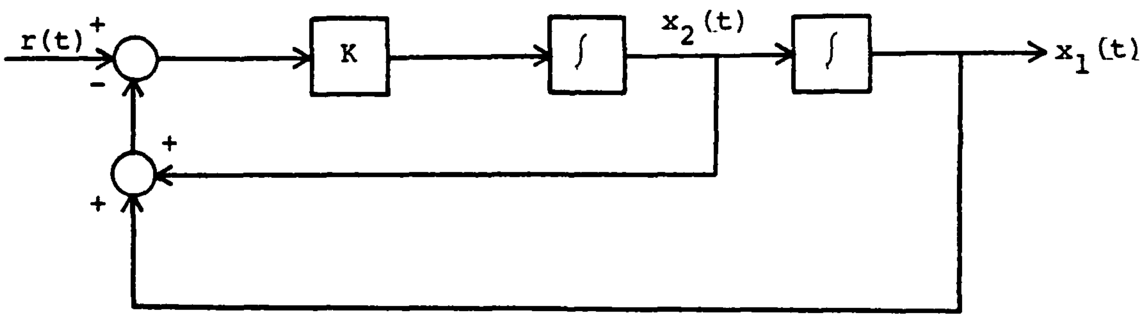


Fig 3-16 Feedback System for GTRESP Test

Since only the time response to initial conditions is required for the problem, r(t) is set equal to zero. The system order is two and

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \underset{\sim}{b}^T = [0 \quad 1], \quad \underset{\sim}{C} = [0 \quad 0]$$

The feedback coefficient matrix $\underset{\sim}{k}^T = [1. \quad 1.]$

The controller gain $K = 1.6$

The initial conditions $\underset{\sim}{x}(0) = [10. \quad 0]$

From the dynamics of the system, a final time of 10 seconds is chosen.

An integration step size of 0.02 is sufficiently small.

The time equations imply that

(1) $$\text{FREQ} \geq \frac{(10 - 0)}{(.02)(100)} \equiv 5$$

(2) 50 points are chosen to cover the ten second interval so

$$\text{FREQ} = \frac{(10 - 0)}{(.02)(50)} = 10$$

(3) One value is going to be plotted every (DT) × (FREQ) or 0.2 second.

(4) The estimated dimensions of the graph can be evaluated as

- dependent variable axis = 36 cm
- independent variable axis = (.318)(50) = 15.9 cm

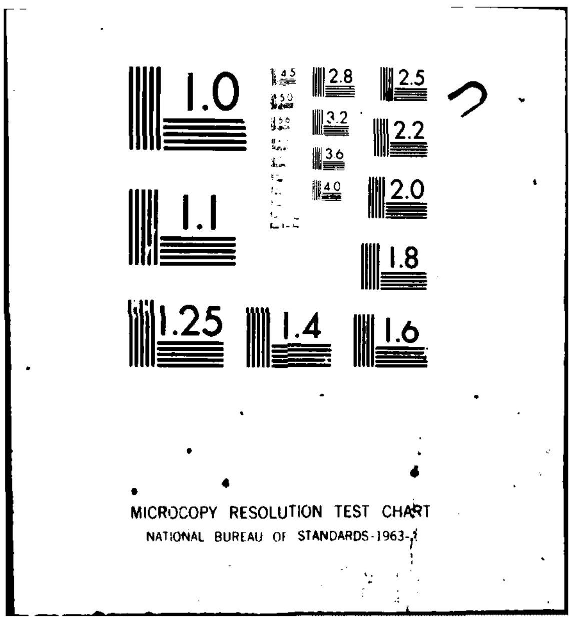
The variables to be plotted are u(t), $x_1(t)$ and $x_2(t)$. All the above are entered as specified on the input format table and the subroutine RFIND(T,R). The complete computer cards set up is then:

```
// (standard OS JOB card),TIME=2
//‸EXEC‸LINCONF
//FORT.SYSIN‸DD‸*
      SUBROUTINE RFIND(T,R)
      R=0.0
      RETURN
      END
/*
//LINK.SYSIN‸DD‸*
‸‸INCLUDE(GTRESP)
‸‸ENTRY‸GTRESP
/*
//GO.SYSIN‸DD‸*
GTRESP TEST     02
0.0    1.0
0.0    0.0
0.0    1.0
0.0    0.0
1.0    1.0
1.6
10.0   0.0
```

AD-A081 770 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A USER'S MANUAL FOR LINEAR CONTROL PROGRAMS ON IBM/360.(U)
DEC 79 B DESJARDINS

UNCLASSIFIED NL

2 OF 3
AD
A081770


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
0.0    10.0    0.02    10.
12U
/*
```

Results are shown in Fig. 3-17A-E

D. MODERN CONTROL SUBPROGRAMS

The following set of subprograms may be used to analyze and design linear feedback control systems which are to achieve a specified closed-loop transfer function.

This group of nine subprograms consists of: the supporting subprograms OBSERV, CONOBS, SENSIT which provide the user with a means of checking the observability and controllability of a system and its sensitivity to parameter variations; the subprograms STVAR, LUEN and SERCOM which help design optimal linear control systems with complete or incomplete state measurements; RICATI and KALMAN which find the feedback and control gains necessary to optimize a given function either for continuous or discrete systems; finally, MIMO which is a computer aided technique to determine feedback control laws for multiple-input multiple-output systems where the number of inputs equals the number of outputs.

The subprograms SENSIT, KALMAN and RICATI are normally Class B subprograms and require a "TIME = 2" specification on the JOB card. All others are Class A. Except for KALMAN which must be operated using Mode Two, all the subprograms are accessed under Mode One or Mode Three.

GRAPHICAL TIME RESPONSE
PROBLEM IDENTIFICATION - GTRESP TEST

****

THE A MATRIX

0.0	1.000000000E 00
0.0	0.0

THE B MATRIX

0.0	1.000000000E 00

THE C MATRIX

0.0	0.0

FEEDBACK COEFF.

1.000000000E 00	1.000000000E 00

GAIN = 1.599999943E 00

INITIAL CONDITIONS

1.000000000E 01	0.0

TZERO = 0.0 TF = 10.000000
DT = 0.020000 FREQ = 10

****

T	Y(T)	U(T)	X1(T)	X2(T)
0.0	0.0	-1.439999E 01	1.000000E 01	0.0
1.999998E-01	0.0	-1.008522E 01	9.741759E 00	-2.438493E 00
3.999996E-01	0.0	-6.402625E 00	9.077955E 00	-4.076353E 00
5.999994E-01	0.0	-3.375700E 00	8.155920E 00	-5.043607E 00
7.999992E-01	0.0	-1.004483E 00	7.096507E 00	-5.471205E 00
9.999990E-01	0.0	7.818846E-01	5.995084E 00	-5.483637E 00
1.199998E 00	0.0	2.033217E 00	4.923159E 00	-5.193920E 00
1.399997E 00	0.0	2.831251E 00	3.930048E 00	-4.700587E 00
1.599996E 00	0.0	3.256497E 00	3.050948E 00	-4.086260E 00
1.799994E 00	0.0	3.387875E 00	2.301130E 00	-3.417593E 00
1.999992E 00	0.0	3.258706E 00	1.684095E 00	-2.745787E 00
2.199991E 00	0.0	3.054247E 00	1.194491E 00	-2.108396E 00
2.399989E 00	0.0	2.710335E 00	8.367298E-01	-1.530693E 00
2.599988E 00	0.0	2.317944E 00	5.322073E-01	-1.027794E 00
2.799986E 00	0.0	1.843376E 00	4.201474E-01	-6.066336E-01
2.999985E 00	0.0	1.493517E 00	3.403575E-01	-2.677559E-01
3.199984E 00	0.0	1.116831E 00	3.074234E-01	-7.093020E-03
3.399982E 00	0.0	7.855106E-01	3.264824E-01	1.825733E-01
3.599981E 00	0.0	5.006763E-01	3.767242E-01	3.103530E-01
3.799979E 00	0.0	2.665079E-01	4.471627E-01	3.862324E-01
3.999978E 00	0.0	8.204776E-02	5.284294E-01	4.202906E-01
4.199976E 00	0.0	-5.641105E-02	6.131319E-01	4.221250E-01
4.399975E 00	0.0	-1.538796E-01	6.957147E-01	4.004602E-01
4.599973E 00	0.0	-2.162731E-01	7.722592E-01	3.629115E-01
4.799972E 00	0.0	-2.457575E-01	8.402493E-01	3.153747E-01
4.999970E 00	0.0	-2.605324E-01	8.983225E-01	2.645104E-01
5.199969E 00	0.0	-2.541339E-01	9.463317E-01	2.128021E-01
5.399968E 00	0.0	-2.356427E-01	9.836159E-01	1.636609E-01
5.599966E 00	0.0	-2.093759E-01	1.011799E 00	1.190612E-01
5.799965E 00	0.0	-1.788923E-01	1.031618E 00	8.018959E-02
5.999963E 00	0.0	-1.470116E-01	1.044287E 00	4.759559E-02
6.199962E 00	0.0	-1.156506E-01	1.051071E 00	2.133551E-02
6.399960E 00	0.0	-8.651317E-02	1.053215E 00	1.105759E-03
6.599959E 00	0.0	-6.116359E-02	1.051870E 00	-1.364309E-02
6.799957E 00	0.0	-3.913396E-02	1.048067E 00	-2.360936E-02
6.999956E 00	0.0	-2.100670E-02	1.042640E 00	-2.955727E-02
7.199955E 00	0.0	-6.058437E-03	1.036452E 00	-3.226377E-02
7.399953E 00	0.0	4.057322E-03	1.029933E 00	-3.247350E-02
7.599952E 00	0.0	1.164673E-02	1.023575E 00	-3.085402E-02
7.799950E 00	0.0	1.652197E-02	1.017670E 00	-2.799584E-02
7.999949E 00	0.0	1.516219E-02	1.012418E 00	-2.439417E-02
8.199947E 00	0.0	[illegible]	1.007725E 00	-2.047343E-02
8.399946E 00	0.0	1.557741E-02	1.004232E 00	-1.646829E-02
8.599944E 00	0.0	1.817763E-02	1.001319E 00	-1.267996E-02
8.799943E 00	0.0	1.617016E-02	9.991309E-01	-9.237256E-03
8.999941E 00	0.0	1.282796E-02	9.975914E-01	-6.233916E-03
9.199940E 00	0.0	1.137390E-02	9.966047E-01	-3.713374E-03
9.399938E 00	0.0	8.572425E-03	9.963779E-01	-1.660770E-03
9.599937E 00	0.0	6.735765E-03	9.959007E-01	-1.130502E-04
9.799936E 00	0.0	4.751427E-03	9.959983E-01	1.031525E-03
9.999934E 00	0.0	3.048619E-03	9.962880E-01	1.306616E-03

Figure 3-17A GTRESP Test - Tabular Output

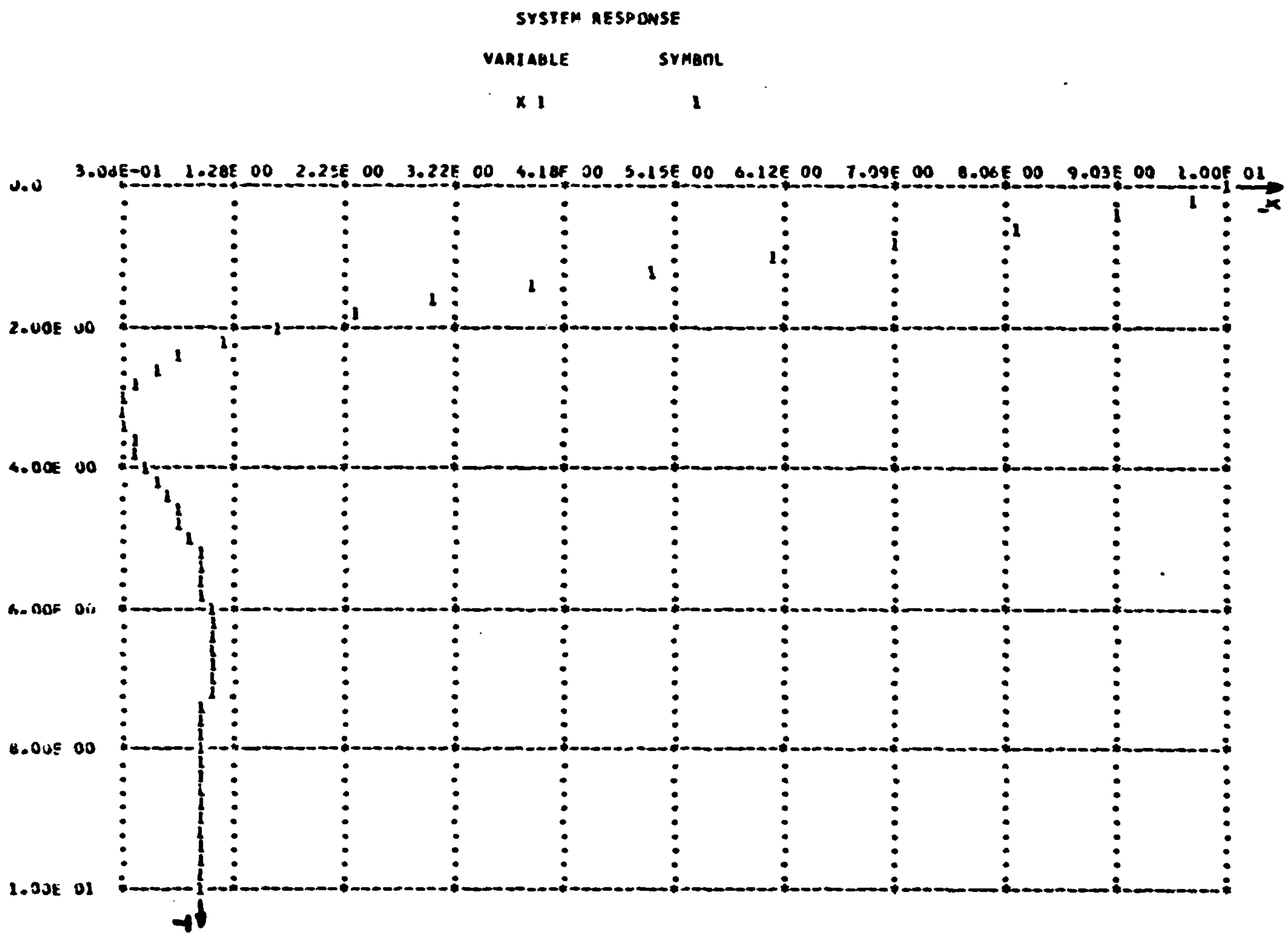


Figure 3-17B System Response - x_1 Versus Time

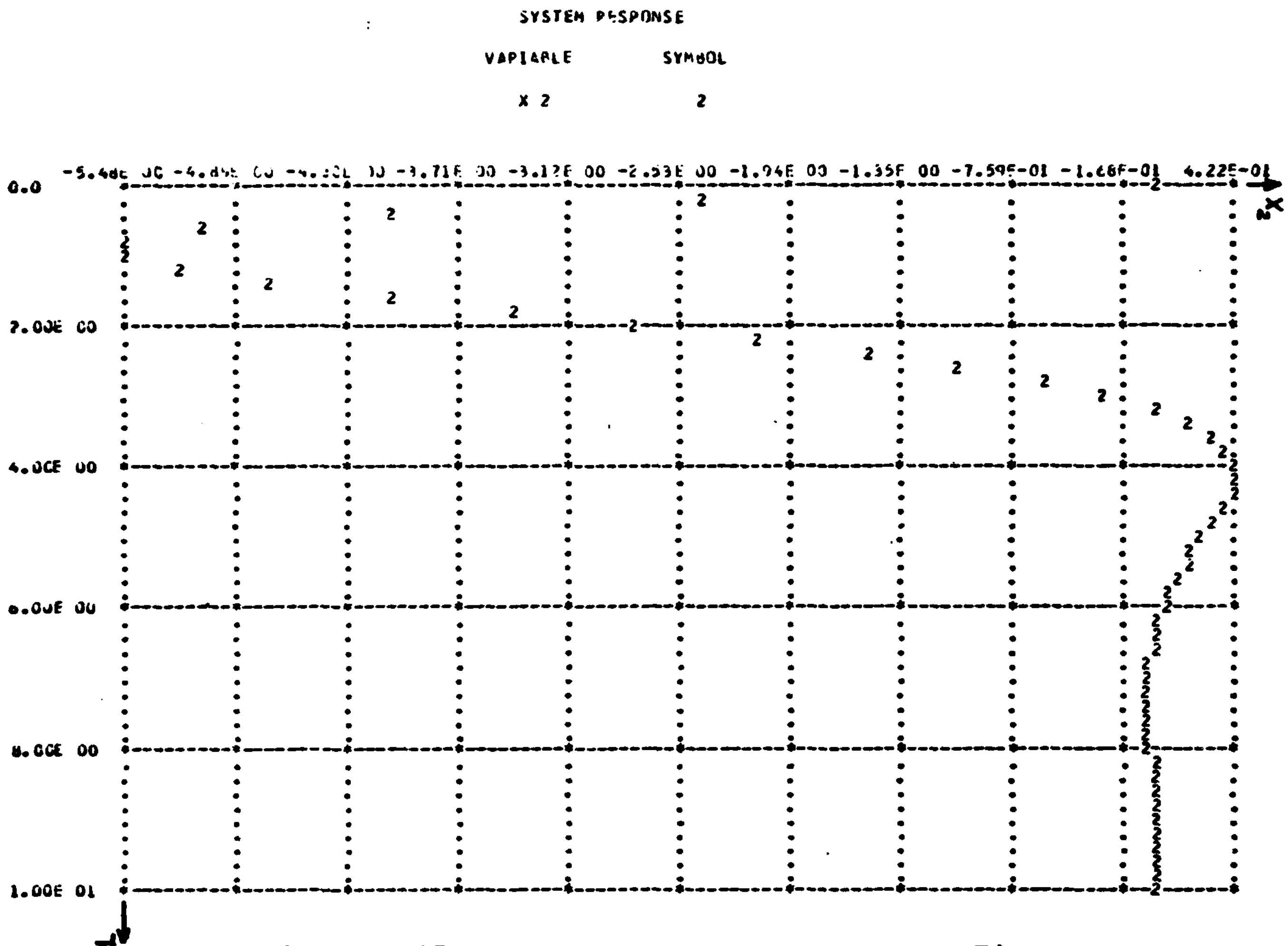


Figure 3-17C System Response - x_2 Versus Time

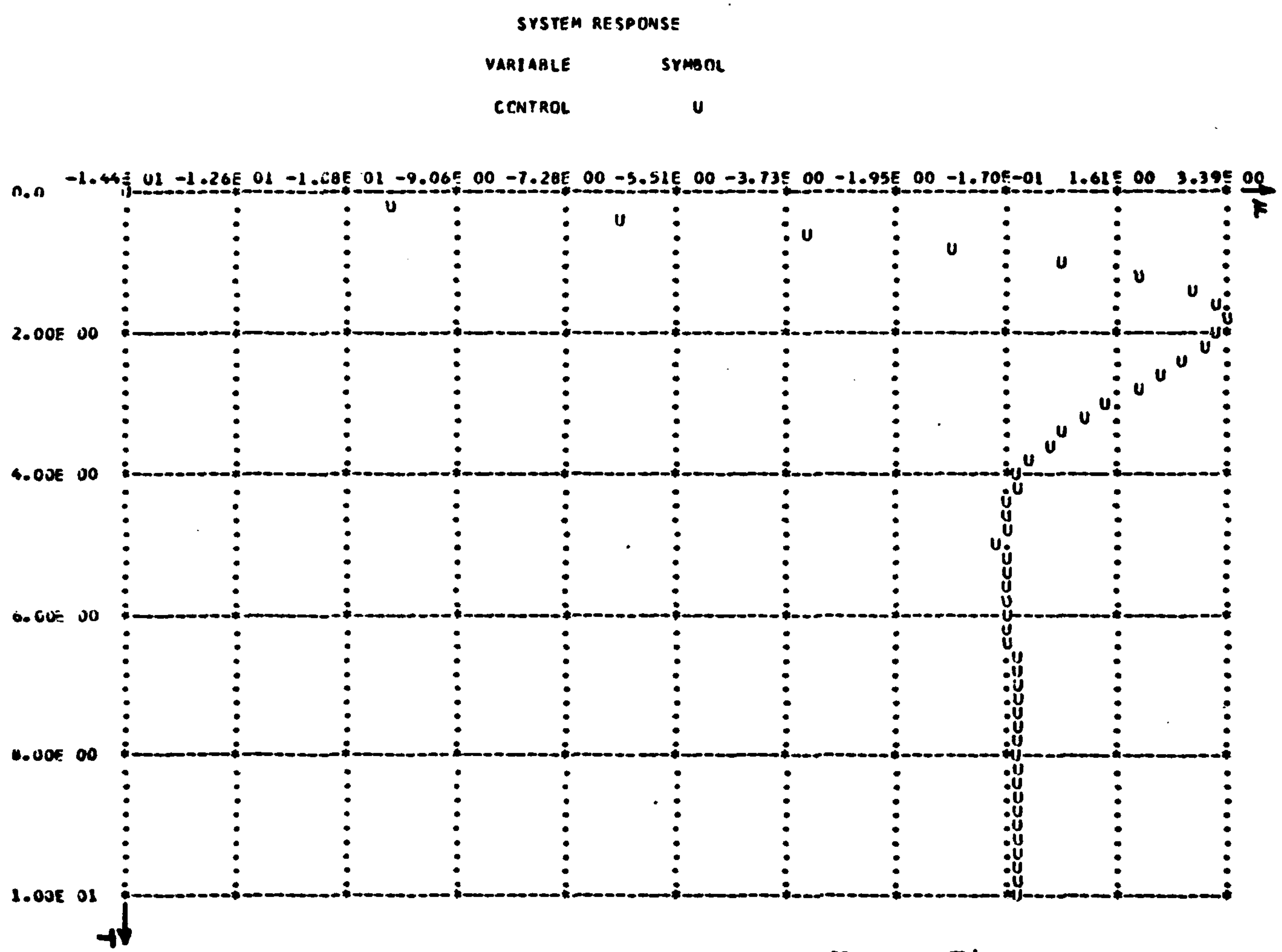


Figure 3-17D System Response - μ Versus Time

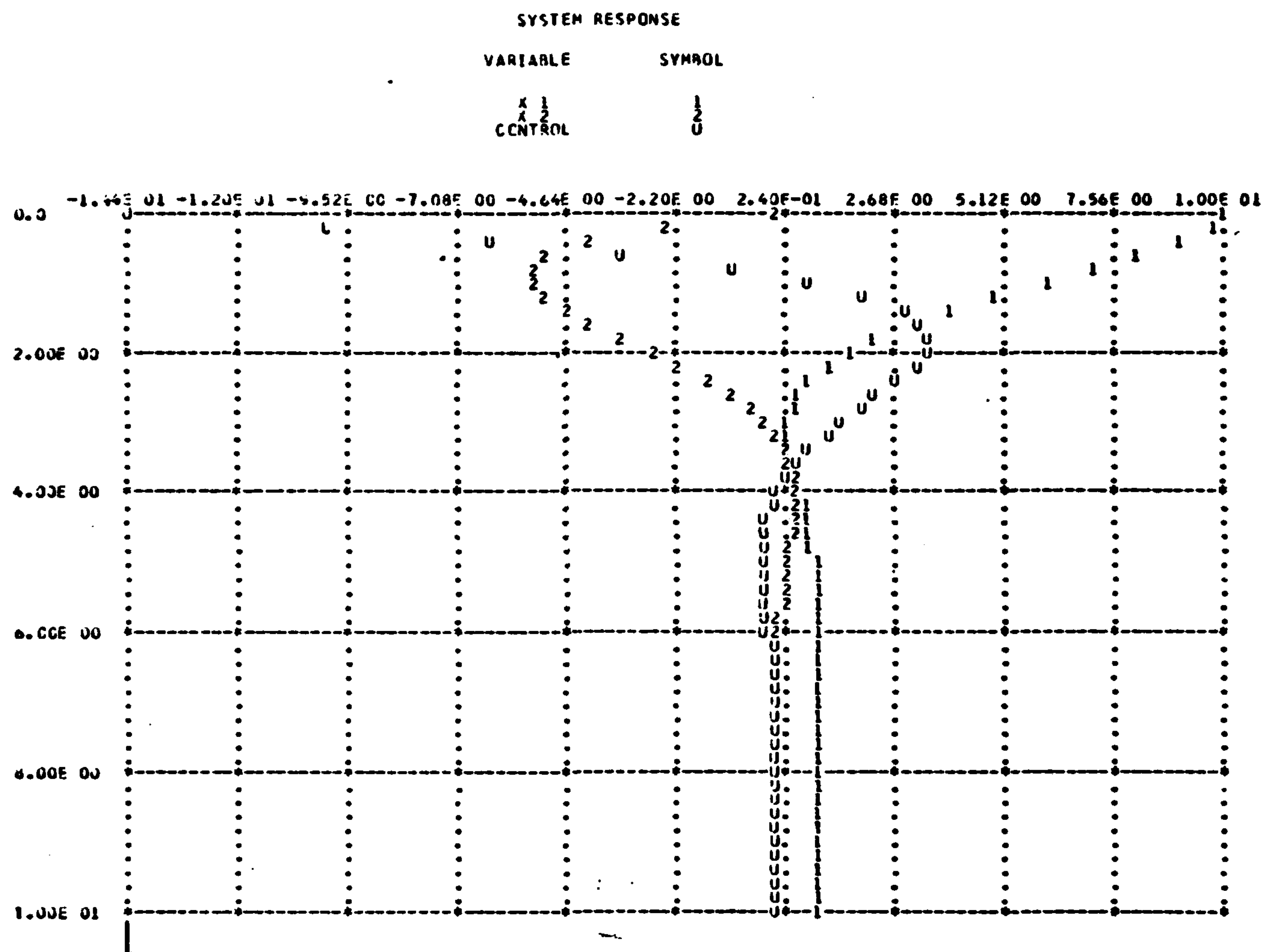


Figure 3-17E System Response - x_1, x_2 and u Versus Time

1. Observability (OBSERV)

This subprogram determines the observability index of the linear, time invariant, Nth order system

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{B}\,\underset{\sim}{u}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

The observability index r of the above system is defined [4] as the smallest positive integer for which the matrix $[\underset{\sim}{C}, \underset{\sim}{A}^T\underset{\sim}{C}, \ldots, (\underset{\sim}{A}^T)^{r-1}\underset{\sim}{C}]$ has rank N.

a. Input

The problem identification, the order of the system and the number of rows of the $\underset{\sim}{C}$ matrix are entered on the first data deck card. Then the $\underset{\sim}{A}$ matrix is presented one row at a time followed by the $\underset{\sim}{C}$ matrix, also one row at a time.

Entry	Input Description	Format	Columns Used
1	Problem identification, order of the system ($N \leq 10$), number of rows of $\underset{\sim}{C}$ ($M \leq 10$)	5A4 2I2	1-20, 21-22, 23-24
2	$\underset{\sim}{A}$ ($N \times N$) matrix (one row per card if $N \leq 8$; one row per two cards if $N > 8$)	8F10.3	1-10, 11-20, etc.
3	$\underset{\sim}{C}$ ($M \times N$) matrix (one row per card if $M \leq 8$; one row per two cards if $M > 8$)	8F10.3	1-10, 11-20, etc.

Table X - Input Format Table for OBSERV

b. Output

The problem identification, and the $\underset{\sim}{A}$ and the $\underset{\sim}{C}$ matrices are listed for reference. Then either "$(\underset{\sim}{A},\underset{\sim}{C})$ is unobservable" is printed or the observability index is given. (If the observability index equals N the number of states, the system is completely observable.)

c. Example

The following set of matrices are to be checked for observability condition.

$$(1) \qquad \underset{\sim}{A} = \begin{bmatrix} -1 & -2 & -2 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \qquad \text{and} \qquad \underset{\sim}{C} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$(2) \qquad \underset{\sim}{A} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix} \qquad \text{and} \qquad \underset{\sim}{C} = \begin{bmatrix} 0 & 1 & 3 \\ 0 & 2 & 4 \end{bmatrix}$$

Here, both (1) and (2) are solved in the same run, placing the data decks one on top of the other giving:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(OBSERV)
/*
//GO.SYSIN^DD^*
OBSERV TEST ONE    0301
```

```
-1.0    -2.0    -2.0
0.0     -1.0    1.0
1.0     0.0     -1.0
1.0     1.0     0.0
OBSERV TEST TWO 0302
2.0     1.0     0.0
0.0     2.0     1.0
0.0     0.0     2.0
0.0     1.0     3.0
0.0     2.0     4.0
/*
```

and the solution is shown in Fig. 3-18.

2. Controllability and Observability (CONOBS)

The subprogram is a modified version of OBSERV. It is used to determine the observability index and check the controllability of a linear, time-invariant control system of the form

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{B}\,\underset{\sim}{u}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

a. Input

The input is the same as for OBSERV except that the $\underset{\sim}{B}^T$ matrix must be included. The input deck starts with the problem identification card which also contains the system order, N, the number of rows of $\underset{\sim}{B}^T$,L, and the number of outputs,

```
OBSERVABILITY INDEX CALCULATION
PROBLEM IDENTIFICATION-     OBSERV TEST ONE

********************************************

THE A MATRIX

-1.0000000E 00      -2.0000000E 00      -2.0000000E 00
 0.0                -1.0000000E 00       1.0000000E 00
 1.0000000E 00       0.0                -1.0000000E 00

THE C MATRIX

 1.0000000E 00       1.0000000E 00       0.0

********************************************

OBSERVABILITY INDEX    3
```

```
OBSERVABILITY INDEX CALCULATION
PROBLEM IDENTIFICATION-     OBSERV TEST TWO

********************************************

THE A MATRIX

 2.0000000E 00       1.0000000E 00       0.0
 0.0                 2.0000000E 00       1.0000000E 00
 0.0                 0.0                 2.0000000E 00

THE C MATRIX

 0.0                 1.0000000E 00       3.0000000E 00
 0.0                 2.0000000E 00       4.0000000E 00

********************************************

(A,C) IS UNOBSERVABLE
```

Figure 3-18 Observability Subprogram Tests

M. Next the $\underset{\sim}{A}$ matrix $(N \times N)$, the $\underset{\sim}{B}^T$ matrix $(L \times N)$ and the $\underset{\sim}{C}$ matrix $(M \times N)$ are entered one row at a time using an 8F10.4 format.

Entry	Input Description	Format	Columns Used
1	Problem identification, system order ($N \leq 10$), number of rows of $\underset{\sim}{B}^T$ ($L \leq 10$), number of outputs ($M \leq 10$)	5A4, 3I2	1-20, 21-22, 23-24, 25-26
2	$\underset{\sim}{A}$ $(N \times N)$ matrix (one row per card if $N \leq 8$; one row per two cards if $N > 8$)	8F10.4	1-10, 11-20, 21-30, etc.
3	$\underset{\sim}{B}^T$ $(L \times N)$ matrix (one row per card if $N \leq 8$; one row per two cards if $N > 8$)	8F10.4	1-10, 11-20, 21-30, etc.
4	$\underset{\sim}{C}$ $(M \times N)$ matrix (one row per card if $N \leq 8$; one row per two cards if $N > 8$)	8F10.4	1-10, 11-20, 21-30, etc.

Table XI - Input Format Table for CONOBS

b. Output

The problem identification and all three matrices are output for reference. Then two sentences are printed indicating whether or not the $(\underset{\sim}{A},\underset{\sim}{C})$ system is observable and the $(\underset{\sim}{A},\underset{\sim}{B})$ system is controllable.

c. Example

The following systems are to be tested for observability and controllability:

$$(1) \quad \dot{\underset{\sim}{x}}(t) = \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 0 \\ -3 & -4 & -2 \end{bmatrix} \underset{\sim}{x}(t) + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \underset{\sim}{u}(t)$$

$$y(t) = x_1(t)$$

$$(2) \quad \dot{\underset{\sim}{x}}(t) = \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ -3 & 0 & -2 \end{bmatrix} \underset{\sim}{x}(t) + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \underset{\sim}{u}(t)$$

$$y(t) = x_1(t)$$

Here again, both solutions are obtained in a single run using one set of control cards before the two consecutive data decks.

For (1),

$$\underset{\sim}{A}(3 \times 3) = \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 0 \\ -3 & -4 & -2 \end{bmatrix}$$

$$\underset{\sim}{B}^T(2 \times 3) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\underset{\sim}{C}(1 \times 3) = [1 \quad 0 \quad 0]$$

Thus system order N = 3, number of rows of $\underset{\sim}{B}^{T}$, L = 2 and the number of outputs M = 1.

For (2),

$$\underset{\sim}{A}(3 \times 3) = \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ -3 & 0 & -2 \end{bmatrix}$$

$$\underset{\sim}{B}(2 \times 3) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\underset{\sim}{C}(1 \times 3) = [1 \quad 0 \quad 0]$$

and the system is of order N = 3, with L = 2 and M = 1.

The control cards and data decks are then as follow:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(CONOBS)
/*
//GO.SYSIN^DD^*
CONOBS TEST ONE  030201
-2.0      0.0      1.0
0.0       -1.0     0.0
-3.0      -4.0     -2.0
0.0       0.0      1.0
```

```
1.0    0.0    0.0
1.0    0.0    0.0
CONOBS TEST TWO 030201
-2.0   0.0    1.0
0.0    -1.0   1.0
-3.0   0.0    -2.0
0.0    0.0    1.0
1.0    0.0    0.0
1.0    0.0    0.0
/*
```

The results are presented in Fig. 3-19.

3. <u>Sensitivity Analysis (SENSIT)</u>

This subprogram is used to obtain the root locus of the closed-loop poles of the (single-input single output) linear control system

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$u(t) = G[r(t) - \underset{\sim}{k}^T\underset{\sim}{x}(t)], \quad \text{where G is a scalar,}$$

as a single element of the plant matrix $\underset{\sim}{A}$, or the control vector $\underset{\sim}{b}$, or the feedback coefficients matrix $\underset{\sim}{k}^T$, or the controller gain K varies between some specified values. As already mentioned, the subprogram studies the effect of a single parameter variation and plots the result. If, for the same system, it is desired to consider more than one parameter variation the user indicates his choices by providing

```
OBSERVABILITY, CONTROLLABILITY
PROBLEM IDENTIFICATION-      CONNORS EXM. ONE
**************************************************
   THE A MATRIX
   -2.0000000E 00        0.0                   1.0000000E 00
    0.0                 -1.0000000E 00         0.0
   -3.0000000E 00       -4.0000000E 00        -2.0000000E 00
   THE B MATRIX
    0.0                  0.0                   1.0000000E 00
    1.0000000E 00        0.0                   0.0
   THE C MATRIX
    1.0000000E 00        0.0                   0.0
**************************************************
   OBSERVABILITY INDEX    3
   (A,B) IS UNCONTROLLABLE
```

```
OBSERVABILITY, CONTROLLABILITY
PROBLEM IDENTIFICATION-      CONNORS EXM. TWO
**************************************************
   THE A MATRIX
   -2.0000000E 00        0.0                   1.0000000E 00
    0.0                 -1.0000000E 00         1.0000000E 00
   -3.0000000E 00        0.0                  -2.0000000E 00
   THE B MATRIX
    0.0                  0.0                   1.0000000E 00
    1.0000000E 00        0.0                   0.0
   THE C MATRIX
    1.0000000E 00        0.0                   0.0
**************************************************
   (A,C) IS UNOBSERVABLE
   (A,B) IS CONTROLLABLE
```

Figure 3-19 Controllability and Observability Subprogram Tests

one option card per element to be varied and the computer completes one root locus for each parameter. The end of a problem is indicated by a blank card. After that card, a data deck pertaining to other systems may be included if desired.

Execution times for this subprogram are normally more than 20 seconds. Thus TIME = 2 should be specified on the JOB card. Mode One or Mode Three is to be used to access the subroutines.

a. Input

The problem identification and the system order ($N \leq 10$) are presented on the first card. Next the plant matrix $\underset{\sim}{A}$ $(N \times N)$ and the $\underset{\sim}{b}^T$ $(1 \times N)$ matrix are entered, followed by the feedback coefficients k_1, k_2, ..., k_n, and the controller gain G. Then the option card is given, indicating the element to be varied, the number of parameter values to be used, and the minimum and the maximum values of that parameter. This card, with the proper modification, is repeated once for each element to be varied. Finally a blank card indicates the end of the problem.

Entry	Input Description	Format	Columns Used
1	Problem identification, order of the system ($N \leq 10$)	5A4 I2	1-20, 21-22
2	$\underset{\sim}{A}$ ($N \times N$) matrix (one row per card if $N < 8$; one row per two cards if $N > 8$)	8F10.3	1-10, 11-20, etc.
3	b^T ($1 \times N$) matrix (one card if $N \leq 8$; on two cards if $N > 8$)	8F10.3	1-10, 11-20, etc.
4	k^T ($1 \times N$) coefficients matrix (one card if $N \leq 8$; two cards if $N > 8$)	8F10.3	1-10, 11-20, etc.
5	Controller gain G	8F10.3	1-10
6 (repeat this entry once for each parameter which is to be varied)	(1) element to be varied (letter A if the element is part of $\underset{\sim}{A}$, letter B if the element is part of $\underset{\sim}{b}$, K if the element is one of the feedback coefficients, G if the element is the controller gain)	A1	1
	(2) row number of the element if from $\underset{\sim}{A}$, $\underset{\sim}{b}$, or $\underset{\sim}{k}$. Otherwise set equal to 00.	I2,	2-3
	(3) column number of the element if from $\underset{\sim}{A}$. Otherwise set equal to 00.	I2,	4-5
	(4) number of parameter values to be used.	I5,	6-10
	(5) minimum value of the parameter.	F10.3,	11-20
	(6) maximum value of the parameter.	F10.3	21-30
7	blank card (this indicates the end of the problem)	(blank)	(blank)

Table XII - Input Format Table for SENSIT

The user must be very careful while preparing the option cards. The following example can best illustrate the procedure. Suppose it is desired to get the root locus of the poles of a closed-loop system as the parameter a_{24} varies from 0.0 to 100.0. A "number of parameter values to be used" of 20 is selected giving the following option card:

column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
data	A	0	2	0	4	0	0	0	2	0	0	.	0								1	0	0	.

If for the same problem, it is also desired to study the variation of the closed-loop poles as b_3 varies from 0.0 to 100.0 with a "number of parameter values to be used" of 10 and also as G varies from -1600. to -1200. with a "number of parameter values to be used" of 25, then the two added option cards would be:

column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
data	B	0	3	0	0	0	0	0	1	0	0	.	0								1	0	0	.		
data	G	0	0	0	0	0	0	0	2	5	-	1	6	0	0	.					-	1	2	0	0	.

b. Output

The problem identification, the $\underset{\sim}{A}$, $\underset{\sim}{b}^T$, $\underset{\sim}{k}^T$ matrices and the gain value G are listed first. Then the first element to be varied and its minimum and maximum values are printed, followed by each parameter value and the closed-loop poles associated with it. Finally the root locus plot is given for each element to be varied.

Note that the "number of parameter values to be used" should be kept small. Since the values of the roots are calculated and printed for each parameter value, 100 values should be regarded as a practical maximum.

c. Example

The stability of the following system is to be investigated under gain variation and the effect of the non-perfect integrator $(\frac{1}{s+\varepsilon})$ looked at.

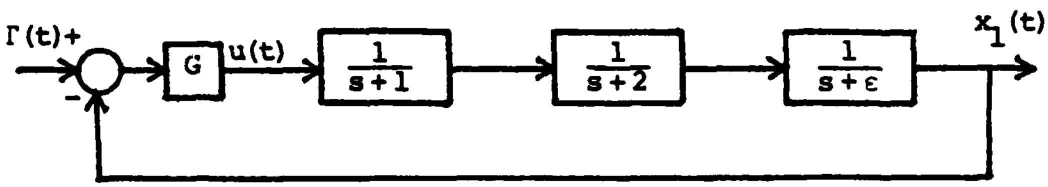


Fig 3-20 Control System for SENSIT Test

First the state equations

$$\dot{x}_1(t) = -\varepsilon x_1(t) + x_2(t)$$

$$\dot{x}_2(t) = -2x_2(t) + x_3(t)$$

$$\dot{x}_3(t) = -x_3(t) + u(t)$$

$$u(t) = G[r(t) - x_1(t)]$$

are written, giving the following data:

Order of the system: N = 3

$$\underset{\sim}{A} = \begin{bmatrix} -\varepsilon & 1 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\underset{\sim}{b}^T = [0 \quad 0 \quad 1]$$

$$\underset{\sim}{k}^T = [1 \quad 0 \quad 0]$$

The gain G is to be varied from 0 to 10 with ε set to zero, and a "number of values to be used" of 20 seems reasonable. The value of ε is also to be varied with G set to its nominal value of 1.0. First ε is set equal to zero in the $\underset{\sim}{A}$ matrix. A range of variation of 0 to 1 and a "number of values to be used" of 25 are selected.

The control and data cards are:

```
// (standard OS JOB card),TIME=2
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(SENSIT)
/*
//GO.SYSIN^DD^*
SENSIT TEST         03
0.0       1.0       0.0
0.0       -2.0      1.0
0.0       0.0       -1.0
```

```
0.0     0.0     1.0
1.0     0.0     0.0
1.0
G0000000200.0   10.0
A010100025  0.0  1.0
(blank card)
/*
```

The results are presented in Fig. 3-20(A-D).

4. State Variable Feedback (STVAR)

This subprogram is a very powerful aid for design and analysis of any single-input single-output linear, time-invariant system represented by the states equations

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$u(t) = K[r(t) - \underset{\sim}{k}^T\,\underset{\sim}{x}(t)]$$

$$y(t) = \underset{\sim}{c}\,\underset{\sim}{x}(t)$$

It permits one to find internal transfer functions of the form $X_i(s)/U(s)$, the plant transfer function $Y(s)/U(s)$, the closed-loop transfer function $Y(s)/R(s)$ and the equivalent feedback transfer function $H_{eq}(s)$. In addition, this subprogram calculates the controller gain and the feedback coefficients necessary to achieve a specified closed-loop transfer function. It is to be run as a Class A job and is accessible under Mode One or Mode Three.

```
SENSITIVITY ANALYSIS PROGRAM
PROBLEM IDENTIFICATION - SENSIT TEST

THE A MATRIX
 0.0                1.0000E 00        0.0
 0.0               -2.0000E 00        1.0000E 00
 0.0                0.0              -1.0000E 00

THE B MATRIX

 0.0                0.0               1.0000E 00

FEEDBACK COEFFICIENTS

 1.0000E 00         0.0               0.0

GAIN =  1.0000E 00

ROOT LOCUS AS THE GAIN VARIES BETWEEN  0.0          AND  1.0000E 01

GAIN =  0.0
ROOTS ARE
REAL PART        IMAG. PART
-1.0000E 00         0.0
-2.0000E 00         0.0
 0.0                0.0

GAIN =  5.263157E-01
ROOTS ARE
REAL PART        IMAG. PART
-4.3025E-01        -2.8122E-01
-4.3025E-01         2.8122E-01
-2.1395E 00         0.0

GAIN =  1.052631E 00
ROOTS ARE
REAL PART        IMAG. PART
-3.3154E-01        -5.6354E-01
-3.3154E-01         5.6354E-01
-2.3369E 00         0.0

GAIN =  1.578947E 00
ROOTS ARE
REAL PART        IMAG. PART
-2.7666E-01        -7.5419E-01
-2.7666E-01         7.5419E-01
-2.4463E 00         0.0

GAIN =  2.105263E 00
ROOTS ARE
REAL PART        IMAG. PART
-2.5369E 00         0.0
-2.3057E-01        -8.8094E-01
-2.3057E-01         8.8094E-01

GAIN =  2.631579E 00
ROOTS ARE
REAL PART        IMAG. PART
-2.6200E 00         0.0
-1.9300E-01        -9.8403E-01
-1.9300E-01         9.8403E-01

GAIN =  3.157894E 00
ROOTS ARE
REAL PART        IMAG. PART
-2.6924E 00         0.0
-1.5361E-01        -1.0720E 00
-1.5361E-01         1.0720E 00

GAIN =  3.684210E 00

 GAIN =  9.473682E 00
 ROOTS ARE
 REAL PART        IMAG. PART
 -3.2737E 00         0.0
  1.3661E-01        -1.6958E 00
  1.3661E-01         1.6958E 00

 GAIN =  9.999998E 00
 ROOTS ARE
 REAL PART        IMAG. PART
 -3.3069E 00         0.0
  1.5445E-01        -1.7316E 00
  1.5445E-01         1.7316E 00

 GAIN =  1.052631E 01
 ROOTS ARE
 REAL PART        IMAG. PART
 -3.3435E 00         0.0
  1.7173E-01        -1.7660E 00
  1.7173E-01         1.7660E 00
```

Figure 3-20A Sensitivity Analysis - Variation of Gain

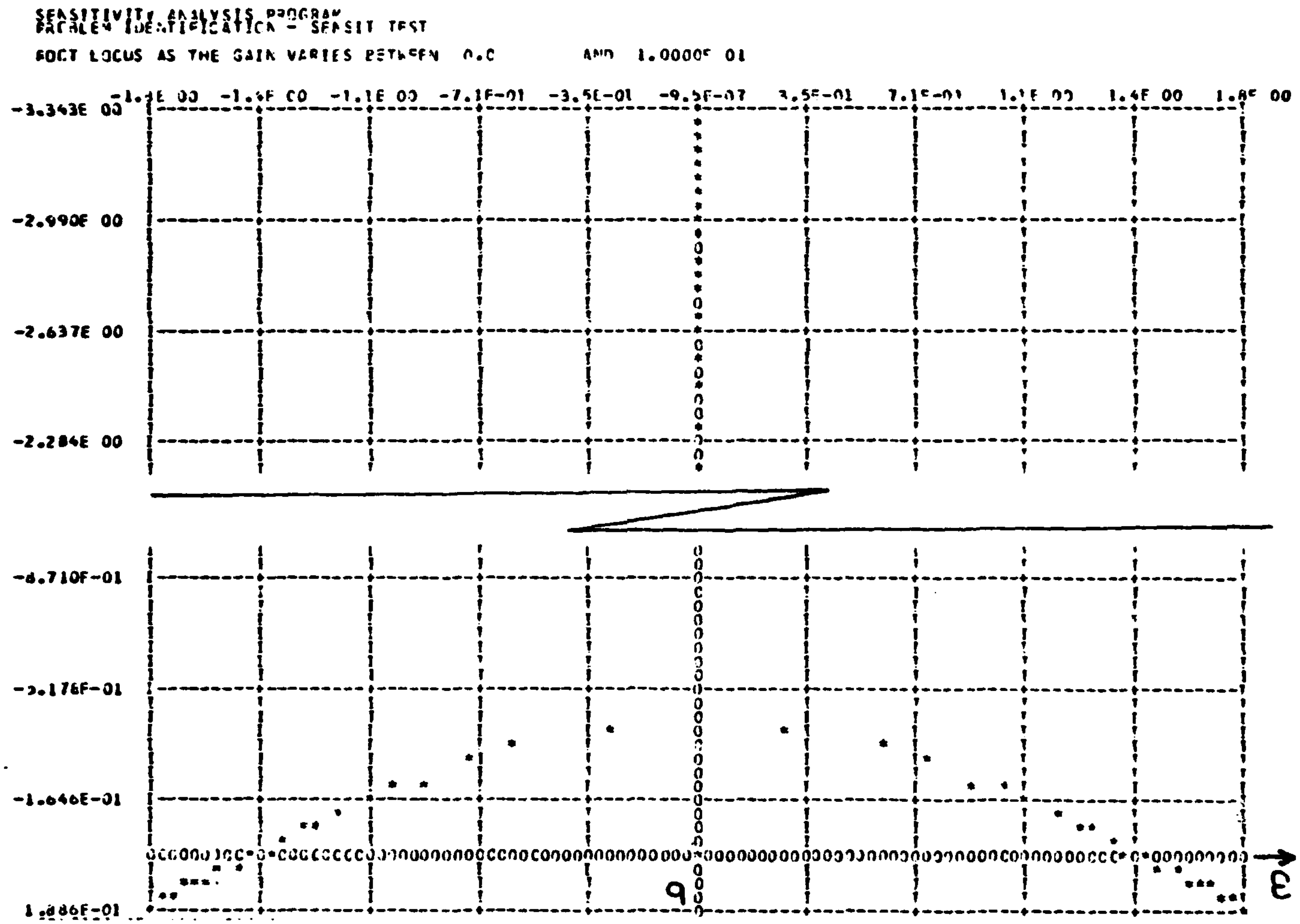


Figure 3-20B Portion of the Root Locus for Variation of Gain

```
SENSITIVITY ANALYSIS PROGRAM
PROBLEM IDENTIFICATION - SENSIT TEST

THE A MATRIX
 0.0              1.0000E 00       0.0
 0.0             -2.0000E 00       1.0000E 00
 0.0              0.0             -1.0000E 00

THE B MATRIX

 0.0              0.0              1.0000E 00

FEEDBACK COEFFICIENTS

 1.0000E 00       0.0              0.0

GAIN =  1.0000E 00

ROOT LOCUS AS A( 1, 1) VARIES BETWEEN  0.0         AND  1.0000E 00

A( 1, 1) =  0.0
ROOTS ARE
REAL PART       IMAG. PART
-2.3247E 00        0.0
-3.3764E-01       -5.6228E-01
-3.3764E-01        5.6228E-01

A( 1, 1) =  4.1667E-02
ROOTS ARE
REAL PART       IMAG. PART
-3.1886E-01       -5.4160E-01
-3.1886E-01        5.4160E-01
-2.3206E 00        0.0

A( 1, 1) =  8.3333E-02
ROOTS ARE
REAL PART       IMAG. PART
-2.3165E 00        0.0
-3.0088E-01       -5.1932E-01
-3.0088E-01        5.1932E-01

A( 1, 1) =  1.2500E-01
ROOTS ARE
REAL PART       IMAG. PART
-2.3126E 00        0.0
-2.8122E-01       -4.9521E-01
-2.8122E-01        4.9521E-01

A( 1, 1) =  1.6667E-01
ROOTS ARE
REAL PART       IMAG. PART
-2.3087E 00        0.0
-2.6232E-01       -4.6899E-01
-2.6232E-01        4.6899E-01

A( 1, 1) =  2.0833E-01
ROOTS ARE
REAL PART       IMAG. PART
-2.3049E 00        0.0
-2.4338E-01       -4.4029E-01
-2.4338E-01        4.4029E-01

A( 1, 1) =  2.5000E-01
ROOTS ARE
REAL PART       IMAG. PART
-2.3012E 00        0.0
-2.2439E-01       -4.0857E-01
-2.2439E-01        4.0857E-01

A( 1, 1) =  9.1667E-01
ROOTS ARE
REAL PART       IMAG. PART
 6.9846E-01        0.0
-2.2521E 00        0.0
-5.2976E-01        0.0

A( 1, 1) =  9.5833E-01
ROOTS ARE
REAL PART       IMAG. PART
 7.5067E-01        0.0
-2.2495E 00        0.0
-5.4285E-01        0.0

A( 1, 1) =  1.0000E 00
ROOTS ARE
REAL PART       IMAG. PART
 8.0194E-01        0.0
-2.2470E 00        0.0
-5.5496E-01        0.0

A( 1, 1) =  1.0417E 00
ROOTS ARE
REAL PART       IMAG. PART
 8.5241E-01        0.0
-2.2445E 00        0.0
-5.6023E-01        0.0
```

Figure 3-20C Sensitivity Analysis - Variation of A(1,1)

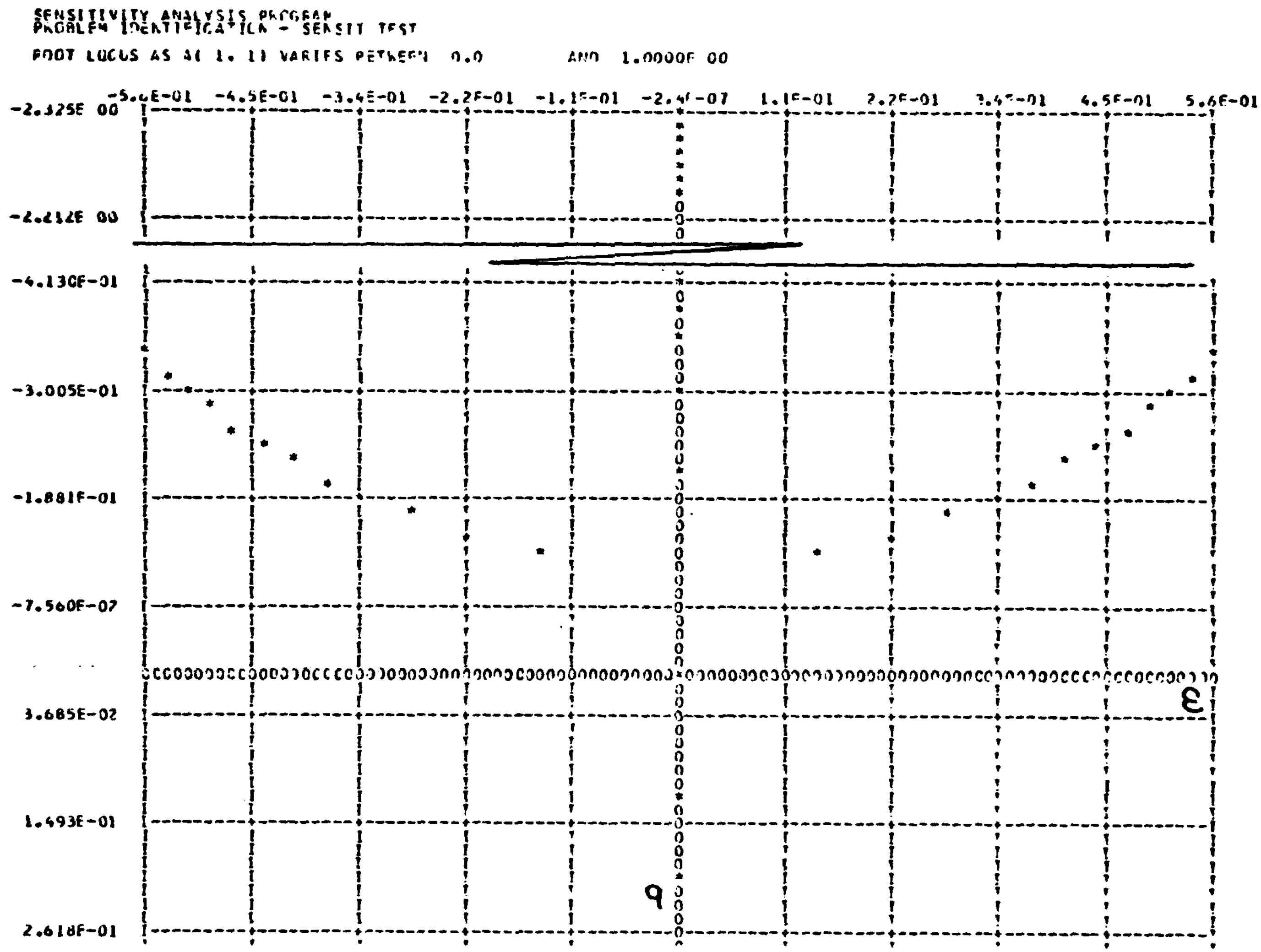


Figure 3-20D Portion of the Root Locus for A(1,1) Variation

The procedure is not very complex, but requires understanding. All the information needed by the user to solve state variable feedback problems is presented in the following paragraphs. However, the theory on which the subprogram is based is not given. The user who wishes to learn more about it should refer to the texts by Schultz and Melsa [5], Melsa and Jones [1], Eveleigh [6] or others.

a. Input

(1) Basic Cards

As usual the problem identification and the system order are given on the first data card, followed by the plant matrix $\underset{\sim}{A}$ $(N \times N)$ and the transpose control vectors $\underset{\sim}{b}^T$ $(1 \times N)$. From this input (which is always required) the subprogram verifies the controllability of the system. Three possible controllability conditions may be found by the computer. One, the system is completely controllable and no special message is printed. Two, the system is numerically uncontrollable. In other words, it is theoretically controllable but uncontrollable in a numerical sense. This situation arises when the controllability matrix

$$\underset{\sim}{E} = [\underset{\sim}{B} \quad \underset{\sim}{AB} \quad \underset{\sim}{AB}^2 \quad \ldots \quad \underset{\sim}{AB}^{n-1}]$$

cannot be accurately inverted using the programmed algorithm. The matrix and its calculated inverse are then multiplied together and checked against the identity matrix to provide a

measure of the uncontrollability of the plant. If the described condition occurs, the message "plant is numerically uncontrollable" is given accompanied by "MAX. DEVIATION=number", where "number" is the value of the deviation from the identity matrix. Reference 1 states that a maximum deviation value larger than 10^{-3} to 10^{-5} has been found to indicate difficulty. The last controllability condition is "the system is uncontrollable", and is indicated as such. Note that even if the plant is determined to be uncontrollable, the computer solves the problem and presents the results. The option of accepting or rejecting the solution is the designer's prerogative.

(2) Open-Loop Cards

The next input cards specify which open-loop transfer functions are to be computed. These cards need not be provided if no internal transfer function is desired. The way to identify the internal transfer functions to be output is by using ficticious $\underset{\sim}{c}_f$ matrices. The following example demonstrates the procedure. Suppose the internal transfer functions $X_2(s)/U(s)$ and $X_1(s)/X_4(s)$ are desired for a fourth order system. Since only $X_i(s)/U(s)$ type of transfer functions are computed by the subprogram, $X_1(s)/U(s)$, $X_2(s)/U(s)$ and $X_4(s)/U(s)$ are requested and the user then only needs to divide $X_1(s)/U(s)$ by $X_4(s)/U(s)$ to obtain $X_1(s)/X_4(s)$. The fictitious $\underset{\sim}{c}_f$ matrices to be provided as input are then:

(1) for $\frac{X_1(s)}{U(s)}$,

$$\underset{\sim}{c}_f = [1 \quad 0 \quad 0 \quad 0]$$

(2) for $\frac{X_2(s)}{U(s)}$,

$$\underset{\sim}{c}_f = [0 \quad 1 \quad 0 \quad 0]$$

(3) for $\frac{X_4(s)}{U(s)}$,

$$\underset{\sim}{c}_f = [0 \quad 0 \quad 0 \quad 1]$$

Following these cards, the real output matrix $\underset{\sim}{c}$ and a null matrix $\underset{\sim}{0}$ $(1 \times N)$ <u>must</u> be entered. The real $\underset{\sim}{c}$ matrix is used to compute $\frac{Y(s)}{U(s)}$ and correctly solve the rest of the problem. The $\underset{\sim}{0}$ matrix is necessary to indicate the end of open-loop calculations.

(3) Closed-Loop Cards

Finally the closed-loop input data are given. Here again the user has a choice among three options.

The first of these closed-loop computations is for analysis only. This choice is indicated by an option card on which the letter A is printed in column one. Following this card, the feedforward gain K and the feedback coefficient matrix $\underset{\sim}{k}^T$ are given as specified on the input format table. From this input, the subprogram determines the closed-loop characteristic polynomial and the numerator of the equivalent feedback transfer function (both the factored and unfactored

forms). From these, the block diagram shown in Fig. 3-21 can be drawn where $G_p(s) = Y(s)/U(s)$ and $H_{eq}(s)$ is the equivalent feedback transfer function.

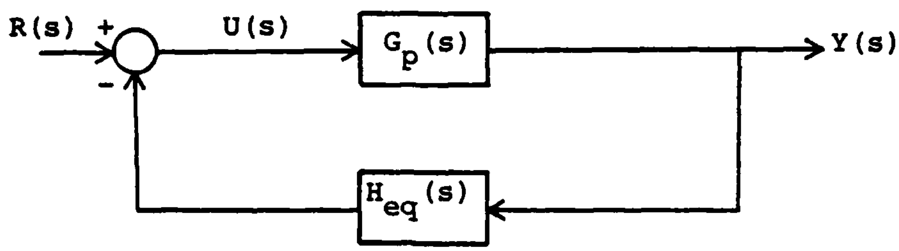


Fig 3-21 H_{eq} Form Block Diagram

The other two closed-loop computations are for design purposes. They are used to calculate the controller gain and the feedback coefficients necessary to achieve a given closed-loop characteristic polynomial. This desired polynomial is the denominator of $Y(s)/R(s)$ and must agree with the system order. If the characteristic polynomial is to be entered in P form, an option card with the letter P in column one is presented followed by one (if $n < 8$) or two (if $n \geq 8$) cards containing the coefficients in ascending order. The coefficient of the highest degree term must always be 1.0 and may be entered as ten blank spaces. On the other hand, if it is more convenient to present it in factored form, the option card has the letter F in the first column and the next cards give the real and imaginary parts of the root using a 2E10.0 format.

Since a user may very well wish to obtain the closed-loop computations for many different characteristic polynomials or try out several values of feedback or feed-forward gains, the subprogram allows one to ask for as many closed-loop computations as desired by placing the input cards one on top of the other.

(4) Problem Termination Card

The last card must be blank. It indicates the end of the problem and must always be present, whether or not the closed-loop portion is included. The following format table conveniently summarizes all the above.

Entry	Input Description	Format	Columns Used
1 (Basic)	Problem identification, system order ($N \leq 10$)	5A4, I2	1-20, 21-22
2 (Basic)	Plant matrix A ($N \times N$) (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, etc.
3 (Basic)	Control vector b^T ($1 \times N$) (one card for $N \leq 8$; two cards for $N > 8$)	8E10.0	1-10, 11-20, etc.
4 (open-loop)	c_f ($1 \times N$) (one card for $N \leq 8$; two cards for $N > 8$) (repeat if several fictitious matrices)	8E10.0	1-10, 11-20, etc.
5 (open-loop)	Output matrix c ($1 \times N$) (one card for $N \leq 8$; two cards for $N > 8$)	8E10.0	1-10, 11-20, etc.
6 (end of open-loop part)	Null matrix 0 ($1 \times N$) (one blank card for $N \leq 8$; two blank cards for $N > 8$)	8E10.0	all

Entry	Input Description	Format	Columns Used
7 Analysis	Letter A in column one	A1	1
8 Analysis	Feedforward gain	8E10.0	1-10
9 Analysis	Feedback coefficient matrix k^T $(1 \times N)$ (on one card for $N \leq 8$; two cards for $N > 8$)	8E10.0	1-10, 11-20, etc.
10 Design option; unfactored form	Letter P in column 1	A1	1
11 Design option; unfactored form	Desired characteristic polynomial coefficients (one card if $N \leq 8$; two cards if $N > 8$). See p. (31) for details	8E10.0	1-10, 11-20, etc.
12 Design option; factored form	Letter F in column 1	A1	1
13 Design option; factored form	Desired characteristic polynomial roots (one per card, real part followed by imaginary part. See p. (32) for details	8E10.0	1-10, 11-20.
14	Blank card (indicates the end of this problem)	8E10.0	blank

Table XIII - Input Format Table for STVAR

Note that entry (4) must be included if no internal transfer function is desired. The same also applies to entries (7-8-9) if analysis option is not desired and (10-11) and/or (12-13) if no design option is taken.

b. Output

The problem identification and the $\underset{\sim}{A}$ and $\underset{\sim}{b}^T$ matrices are given and, if applicable, a numerically or

completely uncontrollable situation is indicated. Next the open-loop calculations are presented. The denominator coefficients in ascending powers of s and the roots of the denominator polynomial are listed at the beginning of the section. Then, if requested, each ficticious $\underset{\sim}{c}_f$ matrix followed by the numerator of the corresponding transfer function is printed. The last output of this section is the $\underset{\sim}{c}$ matrix and the numerator of the plant transfer function. The user is reminded that the ficticious $\underset{\sim}{c}_f$ matrices indicate which $X_i(s)/U(s)$ is computed while the $\underset{\sim}{c}$ matrix specifies the real output y(t) which is used to calculate Y(s)/U(s).

The last section of the printout concerns the closed-loop calculations. If the analysis mode was selected, "KEY=A" is printed followed by the numerator of the equivalent feedback transfer function, $H_{eq}(s)$, both in factored and unfactored forms. Note that the complete $H_{eq}(s)$ is obtained by taking the "numerator of H_{eq}" (given in the closed-loop calculations) and dividing it by the numerator associated with the real $\underset{\sim}{c}$ matrix (given as the last part of the open-loop calculations). Next the feedback coefficients and the gain are listed for reference and the computed closed-loop characteristic polynomial and its roots are given.

If computations of the feedback coefficients and the feedforward gain to achieve a desired closed-loop characteristic polynomial was requested, the computer output shows "KEY=P" or "KEY=F", depending on the design mode

selected. Then, as for the analysis mode, the numerator of the $H_{eq}(s)$ is given, followed by the feedback coefficient matrix $\underset{\sim}{k}^T$ and the feedforward gain K. Here it must be pointed out that the subprogram calculates the gain K so zero steady-state error results from a step input. A designer who wishes other conditions may rescale $\underset{\sim}{K}$ and $\underset{\sim}{k}^T$ appropriately by hand. For example, suppose it is desired to have the controller gain $K = K_1$ but the computer output shows that $K = K_0$ with the feedback coefficients k_1, k_2 and k_3. The procedure is then to modify the results by setting

$$K = K_1$$

and setting

$$\underset{\sim}{k}^T = \frac{K_0}{K_1}[k_1 \quad k_2 \quad k_3]$$

This does not change $Y(s)/R(s)$ and satisfies the condition $K = K_1$. Finally a parameter called "maximum normalized error" is associated with each closed-loop calculation. The value of this parameter indicates the exactitude with which the problem was solved by the computer. This number can help to determine the validity of a solution, especially when numerical uncontrollability was encountered to start with.

c. Example

Eveleigh [6] presents the ideas of design of control systems using state-variable feedback and works out

two examples, the first of which is solved here by the computer method described previously. The problem can be stated as follows: given the plant transfer function

$$G_p(s) = \frac{10}{s(s+1)(s+3)},$$

find each state feedback gain and the feedforward path gain necessary to achieve the closed loop transfer function,

$$G(s) = \frac{10}{s^3 + 4s^2 + 9s + 10}$$

The first step of the procedure is to get the state variable representation of the system. The following signal flow graph may be obtained:

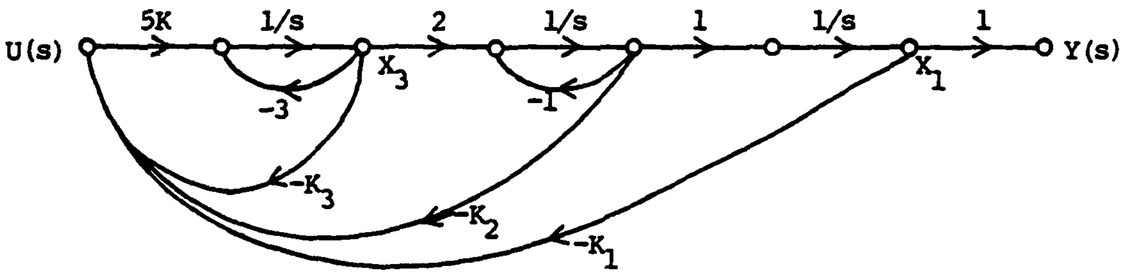


Fig 3-22 Signal Flow Graph for STVAR Test

By inspection,

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 2 \\ 0 & 0 & -3 \end{bmatrix}$$

$$\underset{\sim}{b}^T = [0 \quad 0 \quad 5]$$

$$\underset{\sim}{c} = [1 \quad 0 \quad 0]$$

Normally the user first runs the subprogram for open-loop calculations. Then he either uses it for analysis or for design. To illustrate all the possibilities, the subprogram was applied to solve the same problem using all of the different modes.

(1) Open-Loop Test

For the case at hand, assume the solution is to include the internal transfer function $X_2(s)/U(s)$. Thus the input data requires a ficticious $\underset{\sim}{c}_f$ matrix to be added, i.e.

$$\underset{\sim}{c}_f = [0 \quad 1 \quad 0]$$

The computer card deck for this simple open-loop test is:

```
//^(standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^ ^  INCLUDE^SYSLIB(STVAR)
/*
//GO.SYSIN^DD^*
STVAR OPEN LOOP TEST  03
```

```
0.0     1.0     0.0
0.0    -1.0     2.0
0.0     0.0    -3.0
0.0     0.0     5.0
0.0     1.0     0.0
1.0     0.0     0.0
(blank card)
(blank card)
/*
```

The first blank card is a null matrix $\underset{\sim}{0}$ (1×3) that indicates the end of open-loop calculations while the second blank card indicates the end of the problem. From the results shown in Fig. 3-23,

$$\frac{X_2(s)}{U(s)} = \frac{10s}{s(s+1)(s+3)} = \frac{10}{(s+1)(s+3)}$$

and

$$\frac{Y(s)}{U(s)} = \frac{10}{s(s+1)(s+3)}$$

(2) Analysis Test

To illustrate the analysis computations, the feedforward gain $K = 1$ and the feedback coefficients $k_1 = 1$, $k_2 = 0.6$ and $k_3 = 0$ were assumed. Again the computer card deck is given below.

```
STATE VARIABLE FEEDBACK
PROBLEM IDENTIFICATION -      STVAR OPEN-LOOP TEST
*************************************************
THE A MATRIX
 0.0                  1.0000000E 00      0.0
 0.0                 -1.0000000E 00      2.0000000E 00
 C.0                  0.0               -3.0000000E 00
THE B MATRIX
 0.0                  0.0                5.0000000E 00
*************************************************
OPEN-LOOP CALCULATIONS
DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 0.0                  3.0000000E 00      4.0000000F 00      1.0000000E 00
THE ROOTS ARE               REAL PART          IMAGINARY PART
                         -1.0000000E 00        0.0
                         -3.0000000E 00        0.0
                          C.0                  0.0

THE C MATRIX     *****
 0.0                  1.0000000E 00      0.0
NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 0.0                  1.0000000E 01
THE ROOTS ARE               REAL PART          IMAGINARY PART
                          C.0                  0.0

THE C MATRIX     *****
 1.0000000E 00        0.0                0.0
NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 1.0000000E 01
```

Figure 3-23 State Variable Feedback - Open-Loop Test

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^ ^  INCLUDE SYSLIB(STVAR)
/*
//GO.SYSIN DD *
STVAR ANALYSIS TEST 03
0.0       1.0       0.0
0.0      -1.0       2.0
0.0       0.0      -3.0
0.0       0.0       5.0
1.0       0.0       0.0
(blank card)
A
1.0
1.0       0.6       0.0
(blank card)
/*
```

Interpretation of the output reproduced in Fig 3-24 gives

$$H_{eq}(s) = \frac{10 + 6s}{10} = 1 + .6s$$

and shows that the closed-loop poles are at -2 and $-1 \pm j2$.

(3) Closed-Loop Test

Here the subprogram is used for design. Suppose that the feedforward gain and the feedback coefficient

```
STATE VARIABLE FEEDBACK
PROBLEM IDENTIFICATION -      STVAR ANALYSIS TEST
**************************************************
THE A MATRIX .
 0.0                  1.0000000E 00       0.0
 0.0                 -1.0000000E 00       2.0000000E 00
 0.0                  0.0                -3.0000000E 00
THE B MATRIX
 0.0                  0.0                 5.0000000E 00
**************************************************
OPEN-LOOP CALCULATIONS
DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 0.0                  3.0000000E 00       4.0000000E 00       1.0000000E 00
THE ROOTS ARE                REAL PART            IMAGINARY PART
                           -1.0000000E 00       0.0
                           -3.0000000E 00       0.0
                            0.0                 0.0

THE C MATRIX      *****
 1.0000000E 00        0.0                 0.0
NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 1.0000000E 01
**************************************************
CLOSED-LOOP CALCULATIONS
KEY = A     *****
THE NUMERATOR OF H-EQUIVALENT - IN ASCENDING POWERS OF S
 1.0000000E 01        5.9999990E 00       0.0
THE ROOTS ARE                REAL PART            IMAGINARY PART
                           -1.6666660E 00       0.0
THE FEEDBACK COEFFICIENTS
 1.0000000E 00        5.9999996E-01       0.0
THE GAIN =    1.0000000E 00
THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S
 1.0000000E 01        8.9999981E 00       4.0000000E 00       1.0000000E 00
THE ROOTS ARE                REAL PART            IMAGINARY PART
                           -1.0000000E 00      -2.0000000E 00
                           -1.0000000E 00       2.0000000E 00
                           -2.0000000E 00       0.0
MAXIMUM NORMALIZED ERROR =   1.06E-07
```

Figure 3-24 State Variable Feedback - Analysis Test

values are to be obtained so the closed-loop characteristic polynomial is $s^3 + 4s^2 + 9s + 10$ or, equivalently, the closed-loop poles are located at -2 and $-1 \pm j2$. For illustration, calculations are requested for both the P and the F forms (even though they are exactly the same). The control cards and data deck are then:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^ ^ INCLUDE^SYSLIB(STVAR)
/*
//GO.SYSIN^DD^*
CLOSED LOOP TEST  03
0.0      1.0      0.0
0.0     -1.0      2.0
0.0      0.0     -3.0
0.0      0.0      5.0
1.0      0.0      0.0
(blank card)
F
1.0      2.0
2.0
P
10.0     9.0      4.0      1.0
(blank card)
/*
```

As expected, the results shown in Fig 3-25 specify a gain of one and feedback coefficients values of $k_1 = 1.0$, $k_2 = 0.6$ and $k_3 = 0.0$.

Note that all the above calculations could have been executed as a single run using the following card deck:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^ ^  INCLUDE SYSLIB(STVAR)
/*
//GO.SYSIN^DD^*
STVAR TEST       03
0.0      1.0      0.0
0.0     -1.0      2.0
0.0      0.0     -3.0
0.0      0.0      5.0
0.0      1.0      0.0
1.0      0.0      0.0
(blank card)
F
2.0
1.0      2.0
A
1.0
1.0      0.6      0.0
(blank card)
/*
```

```
STATE VARIABLE FEEDBACK
PROBLEM IDENTIFICATION -      CLOSED-LOOP TEST
*********************************************
THE A MATRIX
 0.0                 1.0000000E 00       0.0
 0.0                -1.0000000E 00       2.0000000E 00
 0.0                 0.0                -3.0000000E 00
THE B MATRIX
 0.0                 0.0                 5.0000000E 00
*********************************************
OPEN-LOOP CALCULATIONS
DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 0.0                 3.0000000E 00       4.0000000E 00       1.0000000E 00
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -1.0000000E 00         0.0
                         -3.0000000E 00         0.0
                          0.0                   0.0

THE C MATRIX     *****
 1.0000000E 00       0.0                 0.0
NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 1.0000000E 01
*********************************************
CLOSED-LOOP CALCULATIONS
KEY = F    *****
THE NUMERATOR OF H-EQUIVALENT - IN ASCENDING POWERS OF S
 1.0000004E 01       6.0000019E 00       0.0
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -1.6666660E 00         0.0
THE FEEDBACK COEFFICIENTS
 1.0000000E 00       6.0000002E-01       0.0
THE GAIN =    9.9999958E-01
THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S
 9.9999962E 00       8.9999962E 00       4.0000000E 00       1.0000000E 00
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -1.0000000E 00        -1.9999990E 00
                         -1.0000000E 00         1.9999990E 00
                         -2.0000000E 00         0.0
MAXIMUM NORMALIZED ERROR =   3.18E-07

KEY = P    *****
THE NUMERATOR OF H-EQUIVALENT - IN ASCENDING POWERS OF S
 1.0000004E 01       6.0000019E 00       0.0
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -1.6666660E 00         0.0
THE FEEDBACK COEFFICIENTS
 1.0000000E 00       6.0000002E-01       0.0
THE GAIN =    9.9999958E-01
THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S
 9.9999962E 00       8.9999962E 00       4.0000000E 00       1.0000000E 00
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -1.0000000E 00        -1.9999990E 00
                         -1.0000000E 00         1.9999990E 00
                         -2.0000000E 00         0.0
MAXIMUM NORMALIZED ERROR =   3.18E-07
```

Figure 3-25 State Variable Feedback - Closed-Loop Test

(4) Step Procedure

The procedure demonstrated through this simple example applies for all problems. The steps to be taken can be summarized as follows:

(a) Obtain the state variable representation of the system.

(b) get $\underset{\sim}{A}$, $\underset{\sim}{b}^T$, and $\underset{\sim}{c}$.

(c) If necessary, define ficticious $\underset{\sim}{c}_f$ matrices to compute "internal" transfer functions.

(d) For analysis, select the feedforward gain K and the feedback coefficients k_1, k_2, ..., k_n.

(e) For design, select the desired closed-loop characteristic polynomial or poles to be achieved.

5. <u>Luenberger Observers (LUEN)</u>

The subprogram LUEN is used to design a combined observer-controller to achieve a desired closed-loop transfer function when some of the states are not accessible. The following paragraphs present a detailed description of the computer aided design procedure. However the theory of Luenberger Observers in the design of linear, time-invariant feedback control is not included in the discussion. Users who are not familiar with the subject should consult references 4 and 7, or any other relevant textbook before working with this subprogram.

The solution plan is to start from the state variable representation of a linear time-invariant system and reconstruct the missing states using an observer. Then, using

both measured and estimated states, assign the feedback coefficients and gains required to properly control the system. The block diagram presented in Fig 3-26 best shows what is meant. The plant represented by the state variable equations

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

must be controllable and observable. Notice that the $\underset{\sim}{C}$ matrix indicates which state variables are measured. For example, a fourth order system with only the states x_2 and x_3 being accessible would yield

$$\underset{\sim}{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The real output to be controlled, denoted by $y_c(t)$, may either be one of the state variables or a linear combination of several of them. The user is to define a desired closed-loop transfer function and find what feedback gains would normally have to be used to obtain it, assuming all states were available. This is done using the subprogram STVAR as explained later in the design procedure.

The subprogram LUEN is then used to calculate all the elements necessary to construct the observer and the

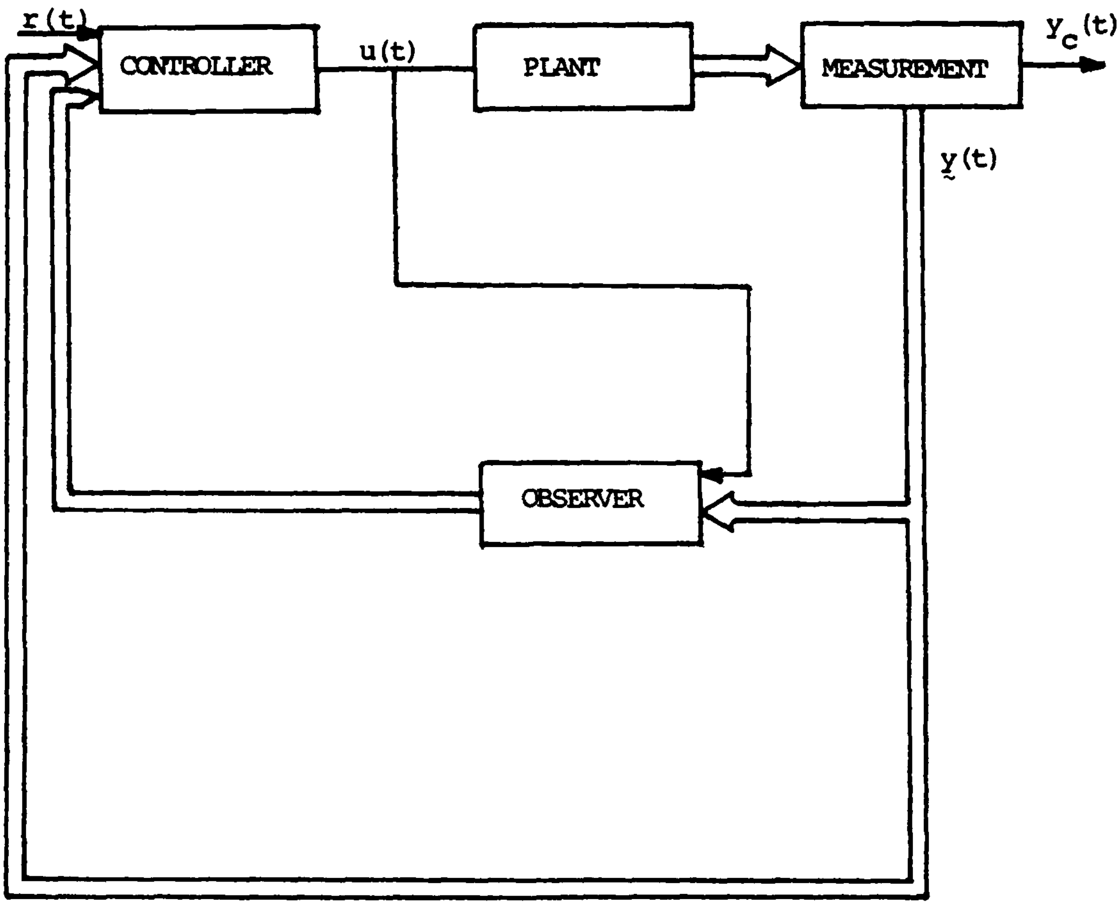


Fig 3-26 Luenberger Observer Block Diagram

controller. The designer only has to specify, in an arbitrary manner, the observer eigenvalues and the necessary feedback coefficients previously found by the use of STVAR. The computer solution gives all the matrices and gains required. Brought together these form the following compensated system:

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$\dot{\hat{\underset{\sim}{x}}}(t) = \underset{\sim}{F}\,\hat{\underset{\sim}{x}}(t) + \underset{\sim}{G}_1\,\underset{\sim}{y}(t) + \underset{\sim}{G}_2\,u(t)$$

$$u(t) = K[r(t) - \underset{\sim}{g}^T\underset{\sim}{y}(t) - \underset{\sim}{h}^T\hat{\underset{\sim}{x}}(t)]$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

where

$\underset{\sim}{x}(t)$ = state vector

$u(t)$ = input to the plant

$\underset{\sim}{y}(t)$ = output vector

$r(t)$ = system forcing input

$\hat{\underset{\sim}{x}}(t)$ = estimated state vector

$\underset{\sim}{A}$ = plant matrix $(N \times N)$

$\underset{\sim}{B}$ = distribution matrix $(N \times 1)$

$\underset{\sim}{F}$ = observer eigenvalue matrix

$\underset{\sim}{G}_1, \underset{\sim}{G}_2$ = observer gain matrices

K = controller gain

$\underset{\sim}{g}^T$ = output feedback coefficient matrix

$\underset{\sim}{h}^T$ = observer feedback coefficient matrix

$\underset{\sim}{C}$ = output matrix.

All these elements except for K, which comes from STVAR results, are given as output of the subprogram LUEN. The four equations defining the compensated system can be easily rearranged, as demonstrated in the example which follows, to simulate the system by the use of the subprogram GTRESP.

a. Design Procedure

The step-by-step design procedure presented here contains the essential information to use the program. It also summarizes the Luenberger Observers design concepts.

Step 1

The closed-loop transfer function $Y_c(s)/R(s)$ to be achieved must be selected and, assuming all states to be measured, we solve for the controller gain K and the feedback coefficients $k_1, k_2, \ldots, k_n$. This is done using the state variable feedback subprogram STVAR, which also checks for system controllability. It must be kept in mind that the $\underset{\sim}{c}$ matrix for STVAR is the matrix associated with the real output $y_c(t)$.

Step 2

If an acceptable solution resulted from STVAR, the observability index must next be determined. This can be done by the use of the subprogram OBSERV, or by hand, using

$$\underset{\sim}{G} = [\underset{\sim}{C}^T \quad \underset{\sim}{A}^T\underset{\sim}{C}^T \quad (\underset{\sim}{A}^T)^2\underset{\sim}{C}^T \quad \ldots \quad (\underset{\sim}{A}^T)^{r-1}\underset{\sim}{C}^T]$$

where the observability index r is the minimum integer such that the matrix $\underset{\sim}{G}$ has rank r. If $(\underset{\sim}{A},\underset{\sim}{C})$ is found to be observable, an observer whose order is equal to or greater than (r-1) can be designed.

Step 3

The eigenvalues of the observer are selected arbitrarily. However, to ensure a unique solution will exist, it is necessary to let the eigenvalues of $\underset{\sim}{F}$ be different from those of $\underset{\sim}{A}$. The eigenvalues of $\underset{\sim}{A}$ were previously calculated by STVAR so it should be very easy to choose some appropriate roots for the observer.

Step 4

Using the input format for LUEN, the data are entered and the subprogram executed. The following system is the final result:

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

$$\dot{\hat{x}}(t) = F\hat{x}(t) + G_1 y(t) + G_2 u(t)$$

$$u(t) = K(r(t) - g^T y(t) - h^T\hat{x}(t))$$

Step 5

If desired, the above equations are rearranged using simple, although sometimes laborious, matrix manipulation as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\hat{x}}(t) \end{bmatrix} = A \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} + b\, r(t)$$

$$y_c(t) = c \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}$$

Note that the above augmented system order is equal to the order of the plant, N, plus the order of the observer. The complete system is finally simulated by the use of GTRESP letting k^T equal zero and K equal to unity.

b. Input

As usual the data deck begins with the problem identification card on which the order of the plant, N, the number of measurements M and the order of the observer, (r-1) or greater, also appears. Next, the plant matrix A $(N \times N)$, the distribution vector b^T $(1 \times N)$ and the measured states matrix C $(M \times N)$ are given one row at a time. The feedback coefficient matrix k^T is then entered exactly as given by

the state variable feedback subprogram (STVAR) output. Finally, the observer eigenvalues, which are different from those of the plant, are supplied either in the form of a characteristic polynomial, option P, or as the roots of that polynomial, option F. The option is specified in column one of the first card by writing the letter P or the letter F. If option P is selected, the characteristic polynomial coefficients are given in the usual ascending order fashion, with the highest order coefficient always set equal to 1.0. For example, if the characteristic polynomial of a third order observer is chosen to be $16 + 4s + 5s^2 + s^3$ the last two data deck cards would then be:

```
P
16.0    4.0    5.0    1.0
```

On the other hand, if the roots are to be entered as such, the letter F is written on the option card followed by the observer eigenvalues presented in the usual manner. For example, if the observer poles are -2, -2, -1+j, -1-j the cards would then be

```
F
2.0
2.0
1.0    1.0
```

Note the sign inversion and the fact that only the complex root with the positive imaginary part is entered.

The following input format table summarizes the above.

Entry	Input Description	Format	Columns Used
1	Problem identification, order of the plant $N \leq 10$, dimension of the output vector M, order of the observer L _ (r-1)	5A4, 3I2	1-20, 21-22, 23-24, 25-26
2	Plant matrix $\underset{\sim}{A}$ $(N \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
3	Distribution matrix $\underset{\sim}{b}^T$ $(1 \times N)$ (one card if $N \leq 8$; two cards if $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
4	Measurement matrix $\underset{\sim}{C}$ $(M \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
5	Feedback coefficient matrix $\underset{\sim}{k}^T$ $(1 \times N)$ (on one card if $N \leq 8$; two cards if $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
6	Letter F (if observer eigenvalues are to be entered as roots) or letter P (if observer eigenvalues are to be supplied by giving a characteristic polynomial)	A1	1
7	Entered the observer eigenvalues as specified on the previous card. (If option F, enter the roots' real and imaginary parts; if option P, give the characteristic polynomial coefficients in ascending order).	8F10.3	1-10, 11-20, 21-30, etc.

Table XIV - Input Format Table for LUEN

c. Output

The problem identification followed by the $\underset{\sim}{A}$, $\underset{\sim}{b}^T$ and $\underset{\sim}{C}$ matrices, the desired feedback coefficients and the

observer eigenvalues, both in factored and unfactored form, are presented for reference. The observer and controller elements are printed next as the $\underset{\sim}{F}$ matrix, the $\underset{\sim}{G}_1$ matrix, the $\underset{\sim}{G}_2$ matrix, the output feedback coefficients $\underset{\sim}{g}^T$ and the compensator feedback coefficients $\underset{\sim}{h}^T$.

The complete solution of a problem should also include the results from the subprograms STVAR, OBSERV and, if a simulation is performed, GTRESP.

d. Example

The example presented by Eveleigh [6] pp. 357-360 was slightly rearranged and the state variables x_3 and x_4 were assumed to be inaccessible. The signal flow graph for the uncompensated system is then:

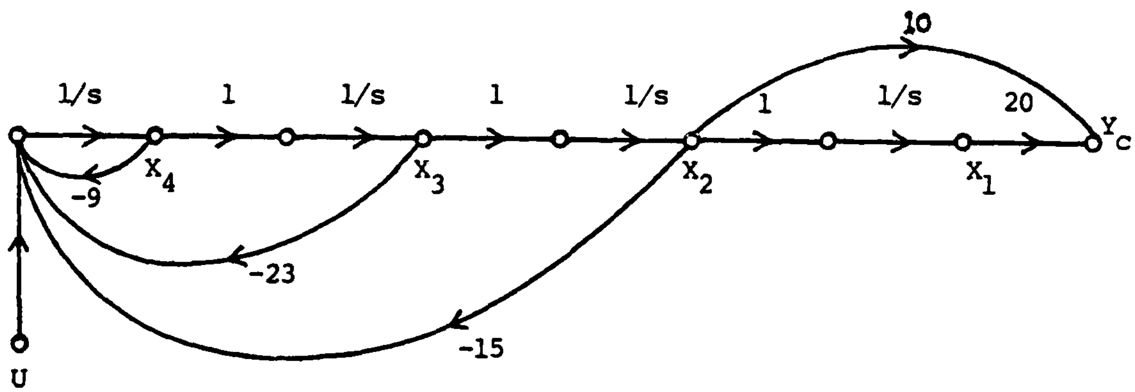


Fig 3-27 Uncompensated System for LUEN Test

where y_c is the controlled output and $x_1(t)$ and $x_2(t)$ are the measured states.

From the diagram, the system matrices are:

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -15 & -23 & -9 \end{bmatrix}$$

$$\underset{\sim}{b}^T = [0 \quad 0 \quad 0 \quad 1]$$

$$\underset{\sim}{c} = [20 \quad 10 \quad 0 \quad 0]$$

The solution presented next utilizes the design procedure of part a.

<u>Step 1</u>

The closed-loop transfer function to be achieved is chosen to be:

$$\frac{Y_c(s)}{R(s)} = \frac{1}{s^4 + 6s^3 + 17s^2 + 28s + 20}$$

$$= \frac{1}{(s+2)(s+2)(s+1+j2)(s+1-j2)}$$

The controller gain K and the feedback coefficients required are found by the use of the subprogram STVAR for which the control cards and data deck are:

```
STATE VARIABLE FEEDBACK
PROBLEM IDENTIFICATION -      STVAR FOR LUEN TEST
*********************************************
THE A MATRIX
0.0              1.0000000E 00       0.0                 0.0
0.0              0.0                 1.0000000E 00       0.0
0.0              0.0                 0.0                 1.0000000E 00
0.0             -1.5000000E 01      -2.3000000E 01      -9.0000000E 00
THE B MATRIX
0.0              0.0                 0.0                 1.0000000E 00
*********************************************
OPEN-LOOP CALCULATIONS
DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
0.0              1.5000000E 01       2.3000000E 01       9.0000000E 00       1.0000000E 00
THE ROOTS ARE          REAL PART           IMAGINARY PART
                      -3.0000000E 00       0.0
                      -4.9999990E 00       0.0
                      -9.9999994E-01       0.0
                       0.0                 0.0

THE C MATRIX     *****
2.0000000E 01    1.0000000E 01       0.0                 0.0
NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
2.0000000E 01    1.0000000E 01
THE ROOTS ARE          REAL PART           IMAGINARY PART
                      -2.0000000E 00       0.0
*********************************************
CLOSED-LOOP CALCULATIONS
KEY = F     *****
THE NUMERATOR OF H-EQUIVALENT - IN ASCENDING POWERS OF S
2.0000000E 01    1.3000000E 01      -6.0000000E 00      -3.0000000E 00
THE ROOTS ARE          REAL PART           IMAGINARY PART
                      -1.0396233E 00       0.0
                       3.2062893E 00       0.0
THE FEEDBACK COEFFICIENTS
2.0000000E 01    1.3000000E 01      -6.0000000E 00      -3.0000000E 00
THE GAIN =    1.0000000E 00
THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S
2.0000000E 01    2.6000000E 01       1.7000000E 01       6.0000000E 00       1.0000000E 00
THE ROOTS ARE          REAL PART           IMAGINARY PART
                      -1.0000000E 00      -2.0000000E 00
                      -1.0000000E 00       2.0000000E 00
                      -2.0004845E 00       0.0
                      -1.9995155E 00       0.0
MAXIMUM NORMALIZED ERROR =   0.0
```

Figure 3-28 STVAR Results for LUEN Test

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(STVAR)
/*
//GO.SYSIN^DD^*
STVAR FOR LUEN TEST 04
0.0      1.0      0.0      0.0
0.0      0.0      1.0      0.0
0.0      0.0      0.0      1.0
0.0     -15.0    -23.0    -9.0
0.0      0.0      0.0      1.0
20.0     10.0     0.0      0.0
(blank card)
F
2.0
2.0
1.0      2.0
(blank card)
/*
```

Results shown in Fig. 3-28 indicate that the system is completely controllable, the plant eigenvalues are -3, -5, -1, and 0, the feedback coefficients are 20, 13, -6 and -3 and the controller gain K equals unity.

Step 2

The observability index is determined using the subprogram OBSERV. The computer cards are as follows:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(OBSERV)
/*
//GO.SYSIN^DD^*
LUEN TEST        0402
0.0      1.0      0.0      0.0
0.0      0.0      1.0      0.0
0.0      0.0      0.0      1.0
0.0      -15.0    -23.0    -9.0
1.0      0.0      0.0      0.0
0.0      1.0      0.0      0.0
/*
```

Step 3

An observability index r = 3 (results taken from OBSERV output, Fig 3-29) permits us to design an observer of order equal to or greater than (r-1) = 2. Here a reduced-order observer is being designed and eigenvalues of -3.5 and -4.0 were selected for the observer. Note that, as required, there are no common eigenvalues for the plant and the observer.

Step 4

The data for the subprogram LUEN are:

system order: 04

number of measurements: 02

order of the observer: 02

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -15 & -23 & -9 \end{bmatrix}, \quad \text{plant matrix}$$

$$\underset{\sim}{b}^T = [0 \quad 0 \quad 0 \quad 1], \quad \text{distribution matrix}$$

$$\underset{\sim}{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \text{state measurement matrix}$$

$$\underset{\sim}{k}^T = [20 \quad 13 \quad -6 \quad -3], \quad \text{desired feedback coefficients (from STVAR)}$$

observer eigenvalues: -3.5, -4.0

The following set of cards is then:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(LUEN)
/*
//GO.SYSIN^DD^*
LUEN TEST     040202
0.0     1.0     0.0     0.0
0.0     0.0     1.0     0.0
0.0     0.0     0.0     1.0
0.0    -15.0   -23.0   -9.0
```

```
0.0     0.0     0.0     1.0
1.0     0.0     0.0     0.0
0.0     1.0     0.0     0.0
20.0    13.0    -6.0    -3.0
F
3.5
4.0
/*
```

From the results shown in Fig 3-30, the complete system can be described as:

$$\dot{\underset{\sim}{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -15 & -23 & -9 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$\hat{\underset{\sim}{x}}(t) = \begin{bmatrix} -7.5 & -1 \\ -14 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_3(t) \\ \hat{x}_4(t) \end{bmatrix} + \begin{bmatrix} 85.5 & 29.25 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} -3 \\ -1.5 \end{bmatrix} u(t)$$

$$u(t) = [1.0]r(t) - [20 \quad 8.5] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} - [1.0 \quad 0.0] \begin{bmatrix} \hat{x}_3(t) \\ \hat{x}_4(t) \end{bmatrix}$$

```
OBSERVABILITY INDEX CALCULATION
PROBLEM IDENTIFICATION-      OBSERV FOR LUEN TEST
**********************************************
THE A MATRIX
 0.0              1.0000000E 00     0.0               0.0
 0.0              0.0               1.0000000E 00     0.0
 0.0              0.0               0.0               1.0000000E 00
 0.0             -1.5000000E 01    -2.3000000E 01    -9.0000000E 00
THE C MATRIX
 1.0000000E 00    0.0               0.0               0.0
 0.0              1.0000000E 00     0.0               0.0
**********************************************
OBSERVABILITY INDEX    3
```

Figure 3-29 OBSERV for LUEN Test

```
LUENBERGER OBSERVER DESIGN PROGRAM
PROBLEM IDENTIFICATION-      LUEN TEST
**********************************************
THE A MATRIX
 0.0              1.0000000E 00     0.0               0.0
 0.0              0.0               1.0000000E 00     0.0
 0.0              0.0               0.0               1.0000000E 00
 0.0             -1.5000000E 01    -2.3000000E 01    -9.0000000E 00
THE B MATRIX
 0.0              0.0               0.0               1.0000000E 00
THE C MATRIX
 1.0000000E 00    0.0               0.0               0.0
 0.0              1.0000000E 00     0.0               0.0
DESIRED FEEDBACK COEFFICIENTS
 2.0000000E 01    1.3000000E 01    -6.0000000E 00    -3.0000000E 00
OBSERVER EIGENVALUES
    REAL PART            IMAG PART
-3.5000000E 00    0.0
-4.0000000E 00    0.0
COEFFICIENTS OF OBSERVER CHARACTERISTIC POLYNOMIAL
-IN ASCENDING POWERS OF S
 1.4000000E 01    7.5000000E 00     1.0000000E 00
**********************************************
THE F MATRIX
-7.5000000E 00    1.0000000E 00
-1.4000000E 01    0.0
THE G1 MATRIX
 8.5499969E 01    2.9249939E 01
 0.0              0.0
THE G2 MATRIX
-2.9999971E 00
-1.5000010E 00
OUTPUT FEEDBACK COEFFICIENTS
 2.0000000E 01    8.5000038E 00
COMPENSATOR FEEDBACK COEFFICIENTS
 1.0000000E 00    0.0
```

Figure 3-30 Luenberger Observer Design - Computer Results

From these equations a block diagram or a signal flow graph could be drawn showing the compensated system.

Step 5

The system is simulated by the use of the graphical time response subprogram (GTRESP) for a unit step input. After some matrix manipulation, the following augmented system is obtained:

$$\begin{bmatrix} \dot{\underset{\sim}{x}}(t) \\ \hline \dot{\hat{\underset{\sim}{x}}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -20 & -23.5 & -23 & -9 & -1 & 0 \\ 145.5 & 54.75 & 0 & 0 & -4.5 & 1 \\ 30 & 12.75 & 0 & 0 & -12.5 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ \hat{x}_3(t) \\ \hat{x}_4(t) \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -3 \\ -1.5 \end{bmatrix} r(t)$$

$$y_c(t) = [20 \quad 10 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \quad \begin{matrix} \underset{\sim}{x}(t) \\ \hat{\underset{\sim}{x}}(t) \end{matrix}$$

$$\underset{\sim}{k}^T = \underset{\sim}{0}$$

$K = 1.0$

For a step input, $r(t) = 1.0$ and initial condition $\begin{bmatrix} \underset{\sim}{x}(0) \\ \hat{\underset{\sim}{x}}(0) \end{bmatrix} = \underset{\sim}{0}$ the initial and final times are 0 and 10, respectively, the integration time step is 0.0025, and the plotting parameter FREQ is 100.

The computer deck for GTRESP is then:

```
// (standard OS JOB card),TIME=2
//^EXEC^LINCONF
//FORT.SYSIN^DD^*
       SUBROUTINE RFIND(T,R)
       R=1.0
       RETURN
       END
/*
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(GTRESP)
/*
//GO.SYSIN^DD^*
GTRESP FOR LUEN TEST  06
0.0        1.0        0.0        0.0        0.0        0.0
0.0        0.0        1.0        0.0        0.0        0.0
0.0        0.0        0.0        1.0        0.0        0.0
-20.0      -23.5      -23.0      -9.0       -1.0       0.0
145.5      54.75      0.0        0.0        -4.5       1.0
30.0       12.75      0.0        0.0        -12.5      0.0
0.0        0.0        0.0        1.0        -3.        -1.5
20.        10.        0.0        0.0        0.0        0.0
```

0.0

1.0

0.0

0.0	10.0	0.002	100.

Y

/*

The results are shown in Fig 3-31. The user is reminded that the observer does supply estimates of the missing components of the state vector but at the expense of adding its own poles to the over-all system.

For comparison, a run is also made simulating the system that would have been obtained if all states were measured, using the feedback coefficients and controller gain from STVAR subprogram results. Since the same forcing input is used, the control cards remain the same and the data cards are changed to read:

ALL STATES MEASURED 04

0.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0
0.0	0.0	0.0	1.0
0.0	-15.0	-23.0	-9.0
0.0	0.0	0.0	1.0
20.0	20.0	0.0	0.0
20.0	13.0	-6.0	-3.0

1.0

(blank card)

0.0	10.0	0.002	100.

Y

/*

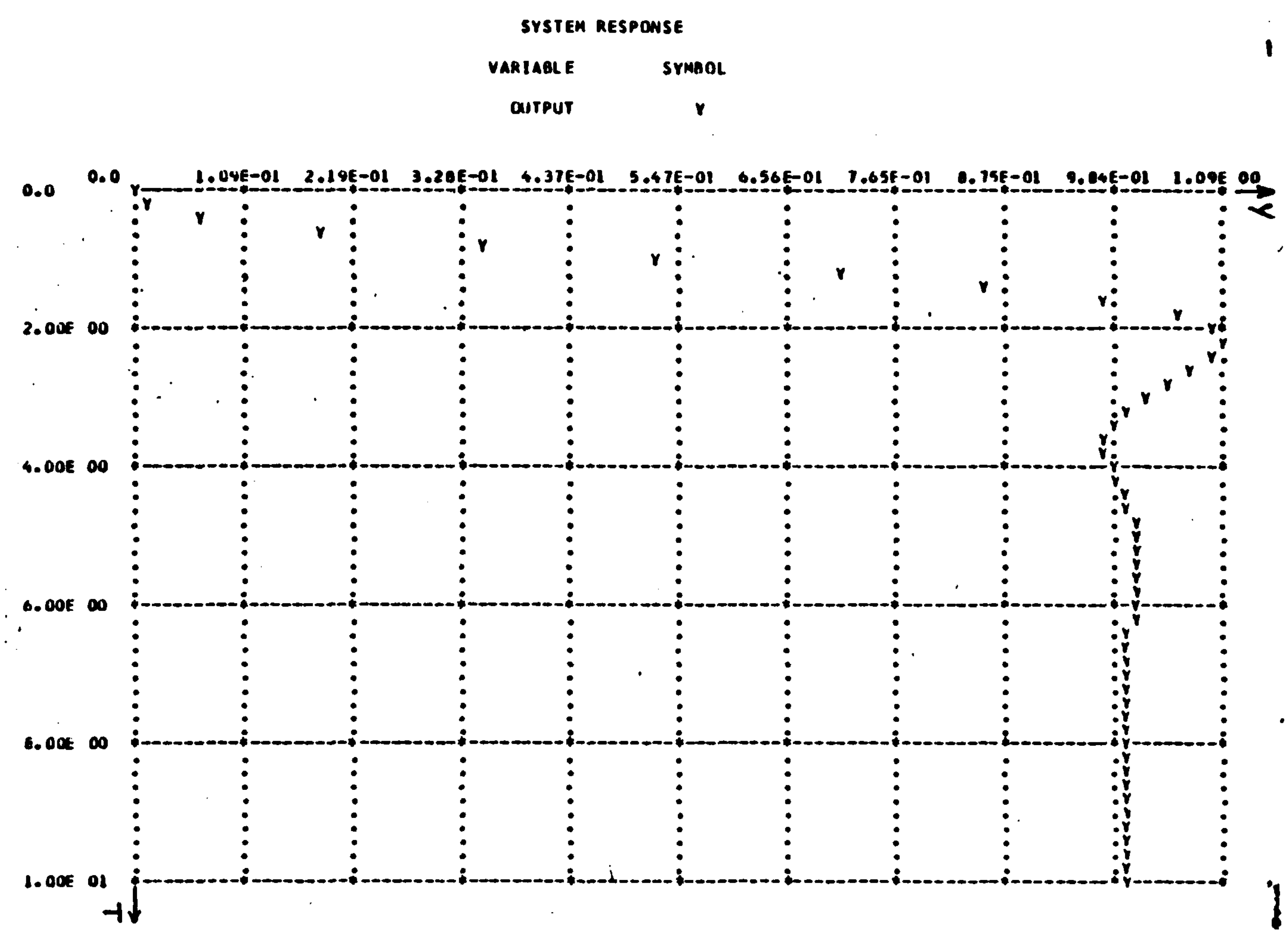


Figure 3-31 GTRESP for Luenberger Observer Test

The time response obtained in Fig 3-32 is almost identical to the one of Fig 3-31, showing that the observer designed does a very good job.

6. Series Compensator (SERCOM)

This subprogram is used to design optimal linear, time-invariant control systems with incomplete state measurements. The optimality criterion here is in terms of a specified closed-loop transfer function to be achieved. The main idea behind the subprogram is to construct a series compensator such that the need for feedback from the unmeasured state variables is eliminated. The way to accomplish this is presented in [8] and [1] and the theory is not repeated here. The user should, however, familiarize himself with the subject before attempting to solve problems by the use of the subprogram SERCOM.

The following paragraphs outline the computer-aided design procedure, the inputs required and the expected output. To illustrate the technique an example problem is worked out in detail. Notice that the overall procedure differs from the one presented in [1].

a. Design Procedure

Before the step-by-step design procedure is outlined, it is necessary to recall the main equations from [8] and [1]. First the uncompensated system state equations are (as for LUFN) of the form

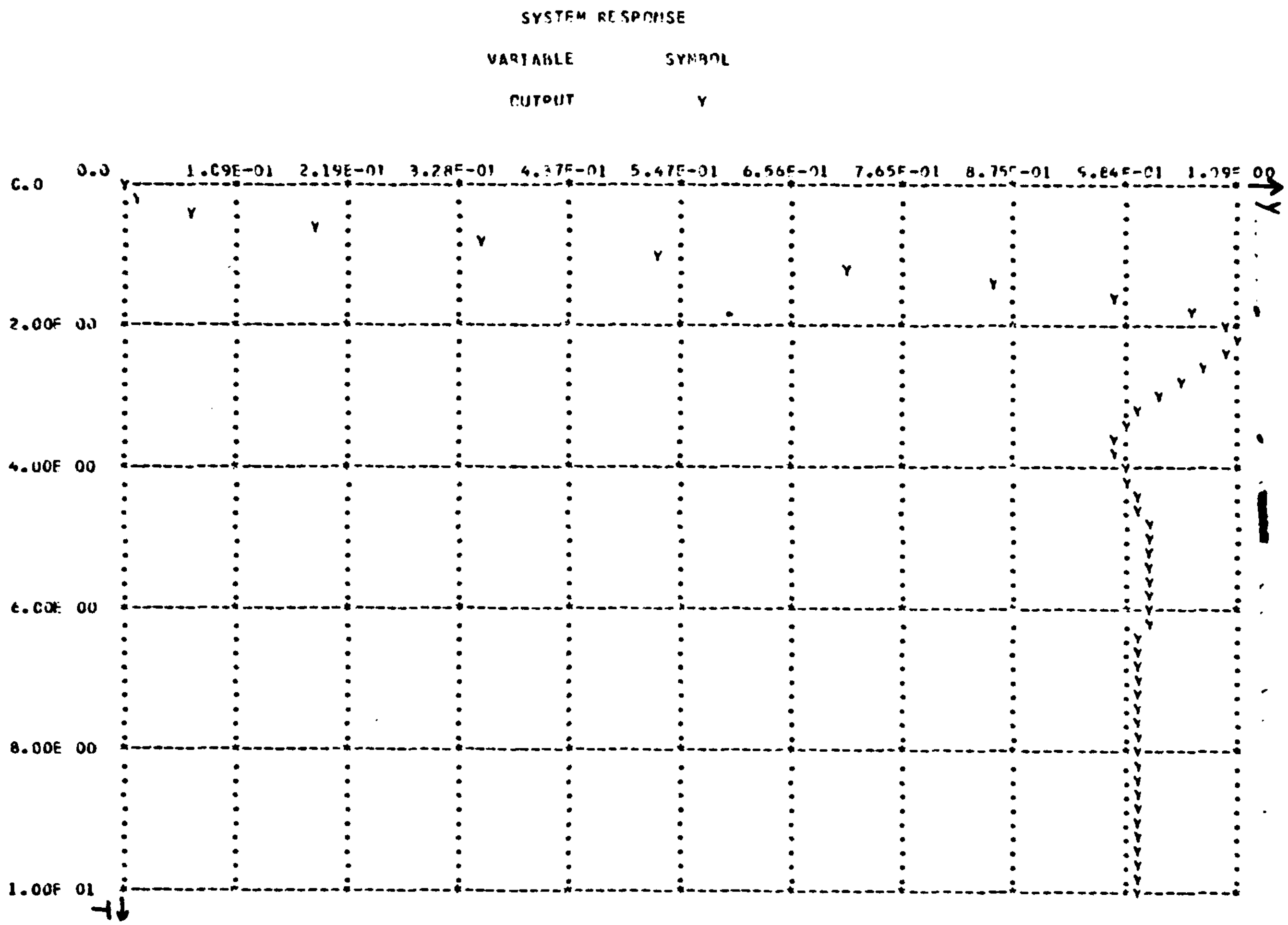


Figure 3-32 GTRESP for Feedback with Complete State Measurements

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{B}\,\underset{\sim}{u}(t)$$

$$y_c(t) = \underset{\sim}{c}\,\underset{\sim}{x}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

where

$y_c(t)$	=	output variable to be controlled (could be one of the measured states or a linear combination of them)
$\underset{\sim}{c}$	=	output variable vector
$\underset{\sim}{y}(t)$	=	vector of measured components of state vector
$\underset{\sim}{C}$	=	state measurement matrix

An arbitrary dynamic controller

$$\dot{\underset{\sim}{z}}(t) = \underset{\sim}{D}\,\underset{\sim}{z}(t) + \underset{\sim}{e}\,w(t)$$

$$u(t) = \underset{\sim}{f}^T\underset{\sim}{z}(t)$$

is added to the above system. It is to be noted that $\underset{\sim}{z}(t)$ are defined as

$$\underset{\sim}{z}^T(t) = [u(t) \quad \dot{u}(t) \quad \ddot{u}(t) \quad \ldots \quad u^{(k-1)}(t)]$$

$$w(t) = u^{(k)}(t)$$

$$\underset{\sim}{f}^T = [1 \quad 0 \quad 0 \quad \ldots \quad 0]$$

$$\underset{\sim}{e} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

and

$$\underset{\sim}{D}^T = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ & & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}$$

The complete system then takes the form

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$\dot{\underset{\sim}{z}}(t) = \underset{\sim}{D}\,\underset{\sim}{z}(t) + \underset{\sim}{e}\,w(t)$$

$$w(t) = \underset{\sim}{f}^T\underset{\sim}{z}(t)$$

$$w(t) = K[r(t) - \underset{\sim}{k}_1{}^T\underset{\sim}{x}(t) - \underset{\sim}{k}_2{}^T\underset{\sim}{z}(t)]$$

The block diagram representation is shown in Figure 3-33A.

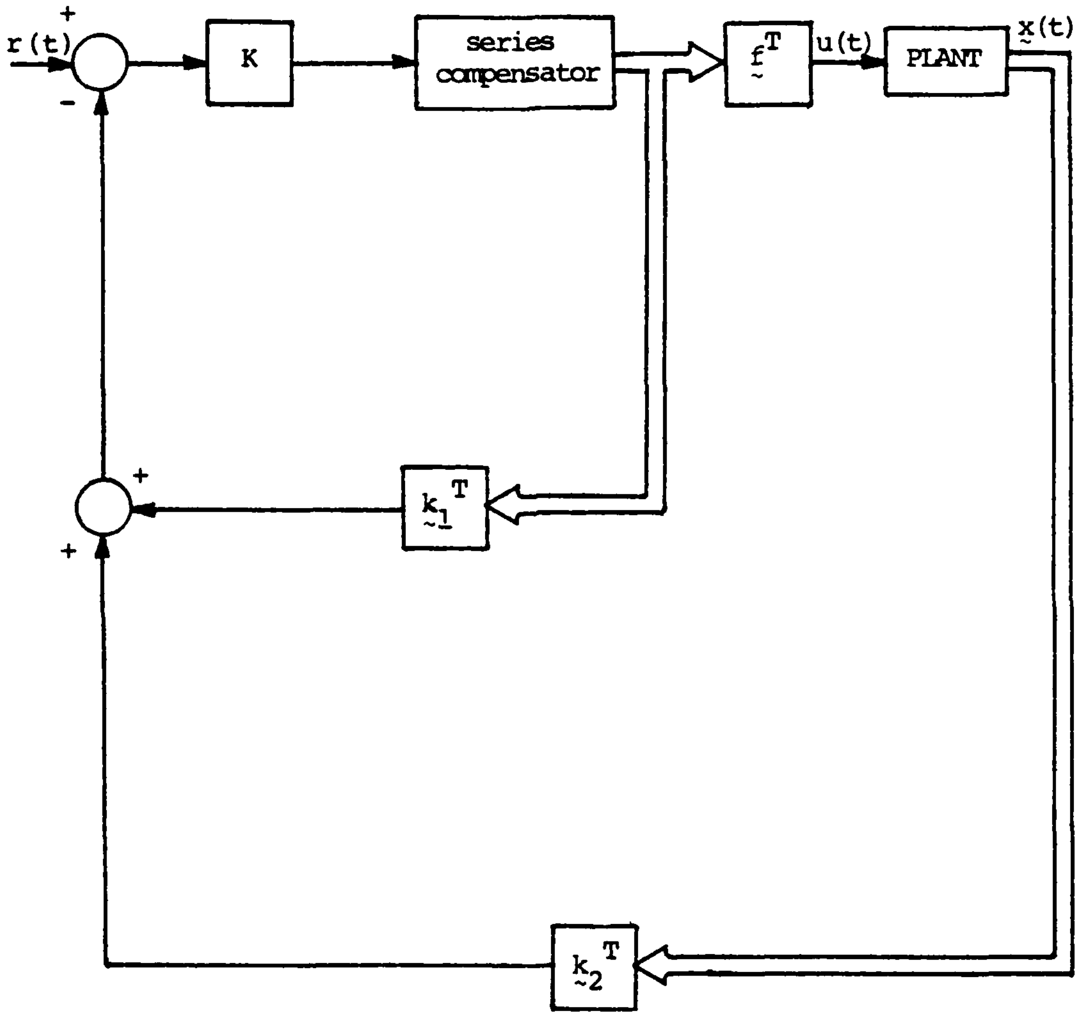


Fig 3-33A Serially Compensated System
with Complete State Measurements

It is clear that this closed-loop system does not solve the problem since it uses all the state variables. It is possible, however, starting from this system, to eliminate the feedback from the unmeasured state variables and this is the purpose of the subprogram SERCOM. Thus, given the above control system, the computer program accomplishes the necessary transformations and outputs the new closed-loop system

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,u(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

$$\dot{\underset{\sim}{v}}(t) = \underset{\sim}{\overline{D}}\,\underset{\sim}{v}(t) + \underset{\sim}{\overline{G}}\,\underset{\sim}{y}(t) + K\underset{\sim}{e}r(t)$$

$$u(t) = \underset{\sim}{f}^T\underset{\sim}{v}(t) + \underset{\sim}{g}^T\underset{\sim}{y}(t)$$

(or in block diagram form, as in Figure 3-33B), with

$\underset{\sim}{A}$ = plant matrix

$\underset{\sim}{b}$ = distribution vector

$\underset{\sim}{\overline{D}}$ = compensator matrix

$\underset{\sim}{\overline{G}}$ = major loop feedback coefficient matrix

$\underset{\sim}{e}$ = input vector

K = input gain (a scalar)

$\underset{\sim}{f}^T$ = compensator output matrix

$\underset{\sim}{g}^T$ = minor loop feedback coefficient matrix

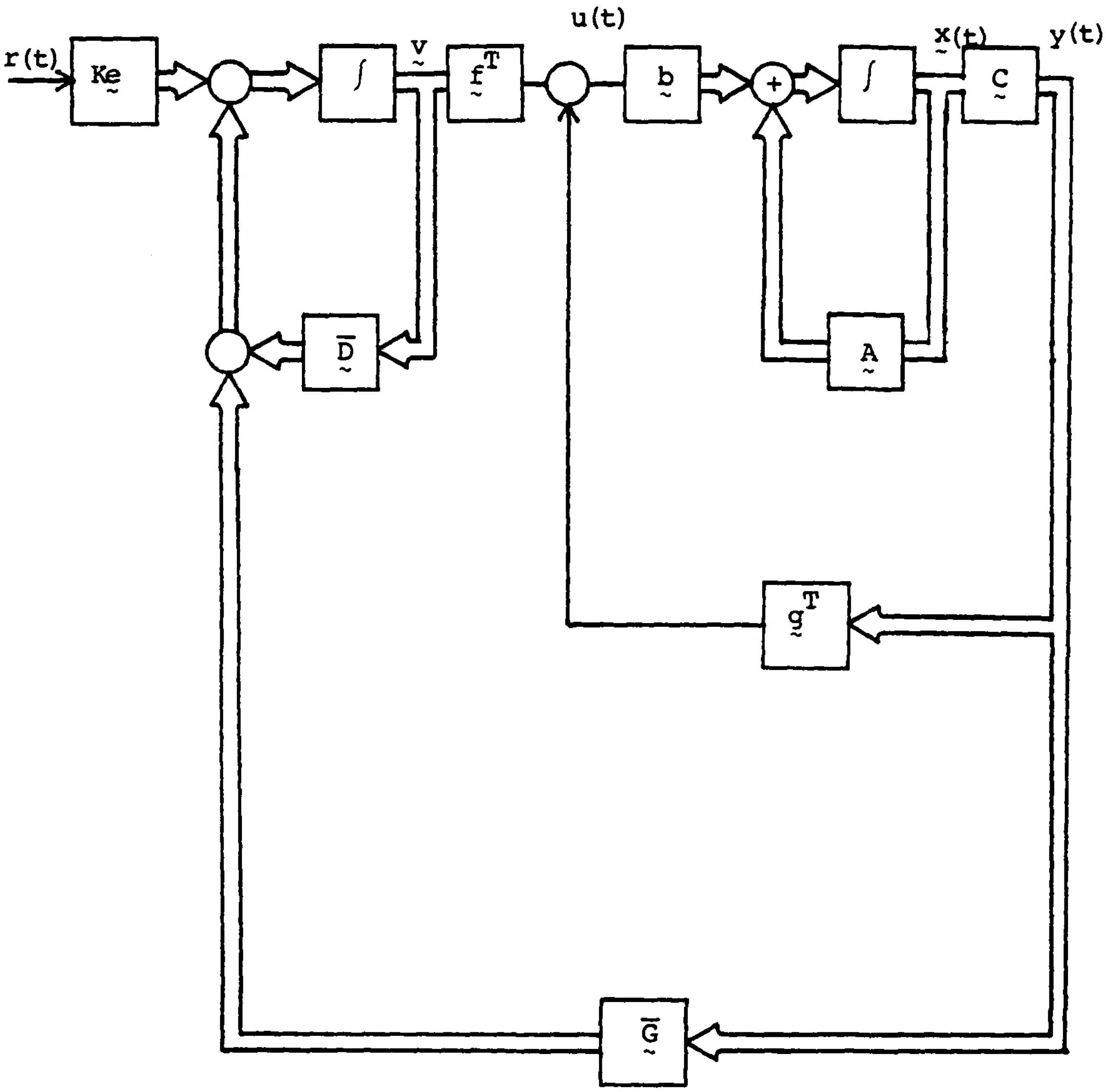


Fig 3-33B Serially Compensated System with Incomplete State Measurements

From theory, such a linear compensator can be designed provided the order of the controller is at least (r-1), where r is the observability index of $(\underset{\sim}{A},\underset{\sim}{C})$ [8].

To summarize the above exposé and give a practical means of using the method, a step-by-step design procedure is presented. After the theory of the series compensator method has been assimilated, it should be sufficient to just follow these few steps and look at the input format table for SERCOM to solve any given problem.

Step 1

The subprogram OBSERV is used to find the observability index r of $(\underset{\sim}{A},\underset{\sim}{C})$. If the system is observable, the minimum order for the compensator is then established as (r-1).

Step 2

The $\underset{\sim}{D}$, $\underset{\sim}{f}^T$ and $\underset{\sim}{e}^T$ matrices are selected such that their dimensions are

$$\underset{\sim}{D}: \quad (r-1) \times (r-1)$$

$$\underset{\sim}{e}^T: \quad 1 \times (r-1)$$

$$\underset{\sim}{f}^T: \quad 1 \times (r-1)$$

It is to be remembered that

$$\underset{\sim}{f}^T = [1 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0]$$

$$\underset{\sim}{e}^T = [0 \quad 0 \quad 0 \quad \ldots \quad 0 \quad 1]$$

$$\underset{\sim}{D} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ & \ldots & \ldots & \ldots & \ldots & \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix}$$

For instance, for a compensator of order one,

$$\underset{\sim}{f}^T = 1$$

$$\underset{\sim}{e}^T = 1$$

$$\underset{\sim}{D} = 0$$

while for a compensator order equal to two,

$$\underset{\sim}{f}^T = [1 \quad 0]$$

$$\underset{\sim}{e}^T = [0 \quad 1]$$

$$\underset{\sim}{D} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

The augmented system is then written as

$$
\begin{bmatrix} \dot{\tilde{x}}(t) \\ \hline \dot{\tilde{z}}(t) \end{bmatrix} = \left[\begin{array}{c|c|c} \tilde{A} & \tilde{b} & \tilde{0} \\ \hline \tilde{0} & \multicolumn{2}{c}{\tilde{D}'} \end{array}\right] \begin{bmatrix} \tilde{x}(t) \\ \tilde{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} u(t)
$$

This form complies with the format necessary to use the subprogram STVAR of step 3.

Step 3

A desired closed-loop transfer function $\frac{Y_c(s)}{R(s)}$ is specified for the augmented system. The order of the combined system is $(n + r - 1)$. For example, suppose that a third order system is to be serially compensated. Its observability index, found using OBSERV, is $r = 2$. Then a fourth order polynomial must be chosen to characterize the desired closed-loop behavior.

At this point, all states are assumed to be available for measurement and the subprogram STVAR is used to obtain the controller gain K and the feedback coefficient matrix $\tilde{k}_1^T$ and $\tilde{k}_2^T$. It is recalled that $\tilde{k}_1^T$ contains the plant feedback coefficients while $\tilde{k}_2^T$ contains those for the compensator.

Step 4

The compensating elements for the augmented system are computed and the required matrix transformations accomplished by the use of the subprogram SERCOM. The final system takes the form

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{b}\,\underset{\sim}{u}(t)$$

$$\dot{\underset{\sim}{v}}(t) = \underset{\sim}{\bar{D}}\,\underset{\sim}{v}(t) + \underset{\sim}{\bar{G}}\,\underset{\sim}{y}(t) + K\underset{\sim}{e}r(t)$$

$$u(t) = \underset{\sim}{f}^T\underset{\sim}{v}(t) + \underset{\sim}{g}^T\underset{\sim}{y}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

where all elements are given in the output of SERCOM.

Step 5

If desired, the compensated system is simulated using GTRESP. As for Luenberger Observers, some simple matrix manipulations are required to put the equations into the form

$$\begin{bmatrix} \dot{\underset{\sim}{x}}(t) \\ \dot{\underset{\sim}{v}}(t) \end{bmatrix} = \underset{\sim}{A}_1 \begin{bmatrix} \underset{\sim}{x}(t) \\ \underset{\sim}{v}(t) \end{bmatrix} + \underset{\sim}{b}_1\, u(t)$$

$$y_c(t) = \underset{\sim}{c}_1 \begin{bmatrix} \underset{\sim}{x}(t) \\ \underset{\sim}{v}(t) \end{bmatrix}$$

$$u(t) = r(t)$$

$$\underset{\sim}{k}^T = \underset{\sim}{0}$$

$$\text{gain} = K$$

$$\underset{\sim}{x}(t_o) = \underset{\sim}{0}$$

The graphical time response subprogram with appropriate time specifications is then run.

b. Input

The data deck includes all the parameters defined for the augmented system. To avoid any mistake, the user should refer to the design procedure for comparison. The input data cards start as usual with the problem identification, the order of the plant N, the number of measurements M and the compensator order (r-1) or greater. The complete system matrices are then presented, one row at a time, in the following order: $\underset{\sim}{A}$ $(N \times N)$, $\underset{\sim}{b}^T$ $(1 \times N)$, $\underset{\sim}{C}$ $(M \times N)$, $\underset{\sim}{D}[(r-1) \times (r-1)]$, $\underset{\sim}{e}^T[1 \times (r-1)]$ and $\underset{\sim}{f}^T[1 \times (r-1)]$. On the final cards, the feedback coefficient matrices $\underset{\sim}{k}_1^T$ and $\underset{\sim}{k}_2^T$ and the controller gain K are presented. For a zero steady-state error to a step input, these would be entered exactly as they appeared on the subprogram STVAR output. The following input format table summarizes the entries required for SERCOM.

Entry	Input Description	Format	Columns Used
1	Problem identification order of the plant ($N \le 10$), number of measurements $= M$, compensator dimension = (r-1) or greater	5A4 3I2	1-20, 21-22, 23-24, 25-26
2	Plant matrix $\underset{\sim}{A}$ $(N \times N)$ (one row per card for $N \le 8$; one row per two cards for $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.

Entry	Input Description	Format	Columns Used
3	Distribution vector b^T $(1 \times N)$ (one card if $N \leq 8$; two cards if $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
4	State measurement matrix C $(M \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
5	Compensator Matrix $D[(r-1) \times (r-1)]$ (one row per card for $(r-1) \leq 8$; one row cards per two cards for $(r-1) > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
6	Input matrix e^T $(1 \times (r-1))$ (one card for $(r-1) \leq 8$; two cards for $(r-1) \geq 8$)	8F10.3	1-10, 11-20, 21-30, etc.
7	Compensator output matrix $f^T[1 \times (r-1)]$ (one card if $(r-1) \leq 8$; two cards if $(r-1) \geq 8$)	8F10.3	1-10, 11-20, 21-30, etc.
8	Feedback coefficients matrix $[k_1^T \;\; k_2^T]$ $(1 \times N+r-1)$ (one card if $(N+r-1) \leq 8$; two cards if $8 \leq (N+r-1) \leq 16$; three cards if $(N+r-1) > 16$)	8F10.3	1-10, 11-20, 21-30, etc.
9	Controller gain K	8F10.3	1-10

Table XV - Input Format Table for SERCOM

c. Output

First the information given as input is listed, i.e., the problem identification, the A, b^T, C, D, e^T, f^T and $[k_1^T \;\; k_2^T]$ matrices and the controller gain K. Next the final compensator system matrix $\bar{D}[(r-1) \times (r-1)]$ is printed (the user must be careful not to confuse this matrix with the original augmented system matrix D), followed by the minor

feedback coefficient matrix $\underset{\sim}{g}^T$ $(1 \times M)$ and the major loop feedback coefficient matrix $\underset{\sim}{\tilde{G}}$ $[(r-1) \times M)$.

d. Examples

Two design examples are worked out. The first one is a simple second order system with only one measured state variable. The other is the fourth order system that was used to demonstrate Luenberger Observers in the previous section.

(1) Example One

A design of a feedback system is required such that the following controllable dynamical equation

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0] \, \underset{\sim}{x}(t)$$

has a time response to a step input approximately the same as for a second order system with poles at $-1 \pm j$.

<u>Step 1.1</u>

The observability index for the system can easily be found, by hand or by the use of the subprogram OBSERV, to be $r = 2$. Thus a first order compensator is sufficient.

<u>Step 1.2</u>

The $\underset{\sim}{D}$, $\underset{\sim}{f}^T$ and $\underset{\sim}{e}^T$ matrices are selected such that:

$$\underset{\sim}{D} = 0$$

$$\underset{\sim}{e}^T = 1$$

$$\underset{\sim}{f}^T = 1$$

and the augmented system takes the form

$$\begin{bmatrix} \dot{\underset{\sim}{x}}(t) \\ \hline \dot{\underset{\sim}{z}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \underset{\sim}{x}(t) \\ \hline \underset{\sim}{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad u(t)$$

Step 1.3

The degree of the characteristic polynomial is then three. Since the desired response was specified to be similar to a second order system with closed-loop poles at $-1+j$ and $-1-j$, it seems appropriate to select these roots plus a third real root with a large negative value. The subprogram STVAR is then used to calculate the required feedback coefficients and the gain for roots at -10, $-1+j$, $-1-j$. The computer deck for STVAR is

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(STVAR)
/*
//GO.SYSIN^DD^*
```

```
STVAR FOR SERCOM1 03
0.0       1.0       0.0
0.0       -1.0      0.0
0.0       0.0       0.0
0.0       0.0       1.0
1.0       0.0       0.0
(blank card)
F
10.
1.        1.
(blank card)
/*
```

The results are shown in Fig. 3-34.

Step 1.4

Sufficient information is now available to run the subprogram SERCOM. We put together the data:

order of the plant = 02

number of measured states = 01

compensator order = 01

plant matrix $\underset{\sim}{A} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$

distribution vector $\underset{\sim}{b}^T = [0 \quad 1]$

state measurement matrix $\underset{\sim}{C} = [1 \quad 0]$

compensator matrix $\underset{\sim}{D} = 0$

input matrix $\underset{\sim}{e}^T = 1$

compensator output matrix $\underset{\sim}{f}^T = 1$

```
STATE VARIABLE FEEDBACK
PROBLEM IDENTIFICATION -      STVAR FOR SERCOM 1
*********************************************
THE A MATRIX
 0.0                 1.0000000E 00       0.0
 0.0                -1.0000000E 00       1.0000000E 00
 0.0                 0.0                 0.0
THE B MATRIX
 0.0                 0.0                 1.0000000E 00
*********************************************
OPEN-LOOP CALCULATIONS
DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 0.0                 0.0                 1.0000000E 00       1.0000000E 00
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -1.0000000E 00         0.0
                          0.0                   0.0
                          0.0                   0.0

THE C MATRIX      *****
 1.0000000E 00       0.0                 0.0
NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 1.0000000E 00
*********************************************
CLOSED-LOOP CALCULATIONS
KEY = F    *****
THE NUMERATOR OF H-EQUIVALENT - IN ASCENDING POWERS OF S
 1.0000000E 00       1.0999994E 00       5.4999995E-01
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -9.0909135E-01         0.0
THE FEEDBACK COEFFICIENTS
 1.0000000E 00       5.4999995E-01       5.4999995E-01
THE GAIN =   2.0000000E 01
THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S
 2.0000000E 01       2.2000000E 01       1.2000000E 01       1.0000000E 00
THE ROOTS ARE              REAL PART            IMAGINARY PART
                         -1.0000000E 01         0.0
                         -1.0000000E 00        -1.0000000E 00
                         -1.0000000E 00         1.0000000E 00
MAXIMUM NORMALIZED ERROR =  6.94E-07
```

Figure 3-34 STVAR Results for SERCOM Test One

$$[\underset{\sim}{k}_1^T \quad \underset{\sim}{k}_2^T] = [1 \quad 0.55 \quad 0.55], \quad \text{(from STVAR output)}$$

$$K = 20 \quad \text{(from STVAR output)}$$

So the control deck and data cards for SERCOM are:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(SERCOM)
/*
//GO.SYSIN^DD^*
SERCOM TEST ONE  020101
0.0    1.0
0.0    -1.0
0.0    1.0
1.0    0.0
0.0
1.0
1.0
1.0    0.55    0.55
20.0
/*
```

From the results reproduced in Fig. 3-35, it is easy to determine the final system as

$$\dot{\underset{\sim}{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \underset{\sim}{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

```
    SERIES COMPENSATOR DESIGN PROGRAM
    PROBLEM IDENTIFICATION-      SERCOM TEST 1
*********************************************
    THE A MATRIX

     0.0                  1.0000000E 00
     0.0                 -1.0000000E 00

    THE B MATRIX

     0.0                  1.0000000E 00

    THE C MATRIX

     1.0000000E 00        0.0

    THE D MATRIX

     0.0

    THE E MATRIX

     1.0000000E 00

    THE F MATRIX

     1.0000000E 00

    DESIRED FEEDBACK COEFFICIENTS

     1.0000000E 00        5.4999995E-01        5.4999995E-01

    THE GAIN =        2.0000000E 01
*********************************************
    THE COMPENSATOR SYSTEM MATRIX

    -1.0999998E 01

    MINOR LOOP FEEDBACK COEFFICIENTS

    -1.1000000E 01

    MAJOR LOOP FEEDBACK COEFFICIENTS

     1.0100003E 02
*********************************************
```

Figure 3-35 Serial Compensator Design - Test One

$$\dot{v}(t) = -11v(t) + 101y(t) + (20)(1)r(t)$$

$$u(t) = v(t) - 11y(t)$$

$$y(t) = x_1(t)$$

or, equivalently,

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -11x_1(t) - x_2(t) + v(t)$$

$$\dot{v}(t) = 101x_1(t) - 11v(t) + 20r(t)$$

<u>Step 1.5</u>

This last set of equations can be readily used in the subprogram GTRESP to simulate the system forced by a unit step input. From the above equations one gets:

$$\underset{\sim}{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ -11 & -1 & 1 \\ 101 & 0 & -11 \end{bmatrix}$$

$$\underset{\sim}{b}^T = [0 \quad 0 \quad 20]$$

$$\underset{\sim}{c} = [1 \quad 0 \quad 0]$$

$$\underset{\sim}{k}^T = \underset{\sim}{0}$$

$$K = 1.0$$

$$\underset{\sim}{x}(t_o) = \underset{\sim}{0}$$

$$t_0 = 0.0 \qquad t_f = 10.0$$

$$dt = 0.002 \qquad FREQ = 100.$$

We assemble the computer card deck as follows:

```
// (standard OS JOB card),TIME=2
//^ EXEC^LINCONF
//FORT.SYSIN^DD^*
       SUBROUTINE RFIND(T,R)
       R = 1.0
       RETURN
       END
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(GTRESP)
^^ENTRY^GTRESP
/*
//GO.SYSIN^DD^*
GTRESP FOR SFRCOM1   03
0.0       1.0        0.0
-11.      -1.0       1.0
```

```
101     0.0     -11.
0.0     0.0     20.0
1.0     0.0     0.0
0.0     0.0     0.0
1.0
0.0     0.0     0.0
0.0     10.0    0.002    100.
Y
/*
```

The time response shown in Fig. 3-36 can be easily compared with the actual feedback system where both state variables are available (by the use of STVAR and GTRESP) and a decision made regarding the suitability of the compensated system.

Here it is important to note that the method increases the order of the system and adds undesired poles. For this reason it is always wise to simulate (using GTRESP). Another good way to investigate the results is to run the subprogram STVAR in open-loop mode for the same set of equations as for GTRESP. This gives the designer a double check on the accuracy of the solution and verifies the controllability. These ideas are demonstrated in the second example.

(2) Example Two

The same problem presented for the Luenberger Observer example is used here, this time with a series compensator. The fourth order uncompensated system

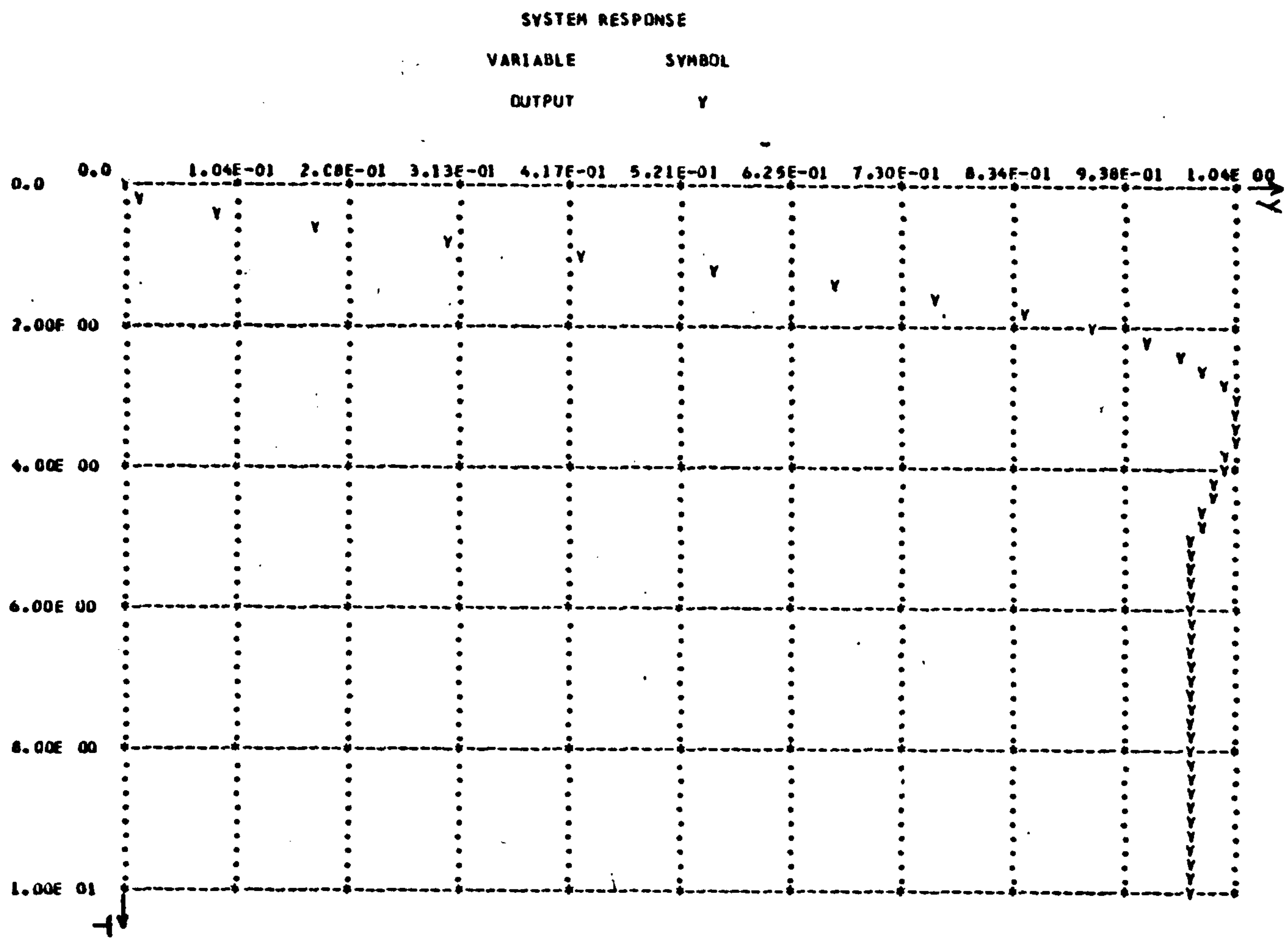


Figure 3-36 GTRESP for SERCOM Test One

$$\dot{\underset{\sim}{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -15 & -23 & -9 \end{bmatrix} \underset{\sim}{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

with measurement equation

$$\underset{\sim}{y}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \underset{\sim}{x}(t)$$

and controlled output

$$y_c(t) = [20 \quad 10 \quad 0 \quad 0] \; \underset{\sim}{x}(t)$$

is to be controlled so the overall time response approaches the one that would result from feeding back the states, if they were all measured, for a fourth order system with closed-loop poles at -2, -2, $-1 \pm j2$.

Step 2.1

The observability index is found by the use of the subprogram OBSERV to be r = 3. Thus the compensator order must be at least (r-1) = 2.

Step 2.2

$\underset{\sim}{D}$, $\underset{\sim}{f}^T$ and $\underset{\sim}{e}^T$ matrices are selected as follow:

$$\underset{\sim}{D} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \underset{\sim}{e}^T = [0 \quad 1], \quad \underset{\sim}{f}^T = [1 \quad 0]$$

and the augmented system becomes

$$\begin{bmatrix} \dot{\underset{\sim}{x}}(t) \\ \dot{\underset{\sim}{z}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -15 & -23 & -9 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \underset{\sim}{x}(t) \\ \underset{\sim}{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y_c(t) = [20 \quad 10 \quad 0 \quad 0 \quad 0 \quad 0] \begin{matrix} \underset{\sim}{x}(t) \\ \underset{\sim}{z}(t) \end{matrix}$$

Step 2.3

Pole placement is usually dictated by some time response specifications. The desired response given here suggests that four of the closed-loop poles be located at -2, -2 and $-1 \pm j2$. The two other roots are undesired and a rule of thumb is to place them to the left of the desired ones. Here -3.5 and -4.0 were selected and the subprograms STVAR run with the following control and data cards:

```
// (standard OS JOB card)
// ^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(STVAR)
/*
```

```
//GO.SYSIN^DD^*
STVAR FOR SERCOM 2  06
0.0       1.0       0.0       0.0       0.0       0.0
0.0       0.0       1.0       0.0       0.0       0.0
0.0       0.0       0.0       1.0       0.0       0.0
0.0       -15.      -23.0     -9.0      1.0       0.0
0.0       0.0       0.0       0.0       0.0       1.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       1.0
20.       10.       0.0       0.0       0.0       0.0
(blank card)
F
1.        2.
2.
2.
3.5
4.0
(blank card)
/*
```

The output shown in Fig. 3-37 gives the gain and feedback coefficients that would be required if all states were measured.

Step 2.4

Since some of the states are not measurable, the subprogram SERCOM is used to transform the original system into the appropriate series compensated system. The information necessary to run the subprogram is:

```
STATE VARIABLE FEEDBACK
PROBLEM IDENTIFICATION -      STVAR FOR SERCOM 2
*********************************************
THE A MATRIX
 0.0              1.0000000E 00     0.0               0.0               0.0               0.0
 0.0              0.0               1.0000000E 00     0.0               0.0               0.0
 0.0              0.0               0.0               1.0000000E 00     0.0               0.0
 0.0             -1.5000000E 01    -2.3000000E 01    -9.0000000E 00     1.0000000E 00     0.0
 0.0              0.0               0.0               0.0               0.0               1.0000000E 00
 0.0              0.0               0.0               0.0               0.0               0.0
THE B MATRIX
 0.0              0.0               0.0               0.0               0.0               1.0000000E 00
*********************************************
OPEN-LOOP CALCULATIONS
DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 0.0              0.0               0.0               1.5000000E 01     2.3000000E 01     9.0000000E 00
 1.0000000E 00
THE ROOTS ARE          REAL PART          IMAGINARY PART
                     -3.0000000E 00       0.0
                     -4.5599990E 00       0.0
                     -5.5999994E-01       0.0
                      0.0                 0.0
                      0.0                 0.0
                      0.0                 0.0

THE C MATRIX     *****
 2.0000000E 01    1.0000000E 01     0.0               0.0               0.0               0.0
NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S
 2.0000000E 01    1.0000000E 01
THE ROOTS ARE          REAL PART          IMAGINARY PART
                     -2.0000000E 00       0.0
*********************************************
```

Figure 3-37 STVAR Results for SERCOM Test Two

```
CLOSED-LOOP CALCULATIONS
KEY = F   *****
THE NUMERATOR OF H-EQUIVALENT - IN ASCENDING POWERS OF S
 2.0000015E 01     3.8714310E 01     3.3428589E 01     1.6035721E 01     3.7857170E 00     3.2142884E-01
THE ROOTS ARE                 REAL PART          IMAGINARY PART
                           -7.8326941E-01      -1.4339800E 00
                           -7.3326941E-01       1.4339800E 00
                           -1.3346539E 00      -4.4440675E-01
                           -1.3346539E 00       4.4440675E-01
THE FEEDBACK COEFFICIENTS
 2.0000015E 01     2.5321426E 01     8.0714178E 00     6.0714012E-01     8.9285684E-01     3.2142884E-01
THE GAIN =   1.3999988E C1
THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL - IN ASCENDING POWERS OF S
 2.7999976E 02     5.4195551E 02     4.6799902E 02     2.3949980E 02     7.5999985E 01     1.3500000E 01
 1.0000000E 00
THE ROOTS ARE                 REAL PART          IMAGINARY PART
                           -5.5994713E-01      -1.9999914E 00
                           -5.5558713E-01       1.9999914E 00
                           -4.0001497E 00       0.0
                           -3.4596652E 00       0.0
                           -1.5683041E 00       0.0
                           -2.0119047E 00       0.0
MAXIMUM NORMALIZED ERROR =  1.57E-06
```

Figure 3-37 (Cont.) STVAR Results for SERCOM Test Two

order of the plant = 04

number of measured states = 02

compensator order = 02

$$\text{plant matrix } \underset{\sim}{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -15 & -23 & -9 \end{bmatrix}$$

distribution vector $\underset{\sim}{b}^T = [0 \quad 0 \quad 0 \quad 1]$

$$\text{state measurement matrix } \underset{\sim}{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{compensator matrix } \underset{\sim}{D} = \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix}$$

input matrix $\underset{\sim}{e}^T = [0 \quad 1]$

compensator output matrix $f^T = [1 \quad 0]$

$[\underset{\sim}{k}_1^T \quad \underset{\sim}{k}_2^T] = [20. \quad 25.3 \quad 8.07 \quad .607 \quad .893 \quad .321]$,
(from STVAR output)

$K = 14.$, from STVAR output.

The computer card deck is then

```
// (standard OS JOB card)
// ^ EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(SERCOM)
/*
//GO.SYSIN^DD^*
SERCOM TEST ONE  040202
```

```
0.0     1.0     0.0     0.0
0.0     0.0     1.0     0.0
0.0     0.0     0.0     1.0
0.0     -15.    -23.    -9.
0.0     0.0     0.0     1.0
1.0     0.0     0.0     0.0
0.0     1.0     0.0     0.0
0.0     1.0
0.0     0.0
0.0     1.0
1.0     0.0
20.     25.321  8.071   .607    .8983  .3214
14.
/*
```

The computer output (Fig. 3-38) gives the compensated system

$$\dot{\underset{\sim}{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -15 & -23 & -9 \end{bmatrix} \underset{\sim}{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$\dot{\underset{\sim}{v}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \underset{\sim}{v}(t) + \begin{bmatrix} 88.75 & -20.74 \\ 0 & 0 \end{bmatrix} \underset{\sim}{y}(t)$$

$$+ \; 14 \, r(t)$$

```
SERIES COMPENSATOR DESIGN PROGRAM
PROBLEM IDENTIFICATION-     SERCOM TEST 2
********************************************
THE A MATRIX

 0.0                  1.0000000E 00        0.0                  0.0
 0.0                  0.0                  1.0000000E 00        0.0
 0.0                  0.0                  0.0                  1.0000000E 00
 0.0                 -1.5000000E 01       -2.3000000E 01       -9.0000000E 00
THE B MATRIX

 0.0                  0.0                  0.0                  1.0000000E 00
THE C MATRIX

 1.0000000E 00        0.0                  0.0                  0.0
 0.0                  1.0000000E 00        0.0                  0.0
THE D MATRIX

 0.0                  1.0000000E 00
 0.0                  0.0
THE E MATRIX

 0.0                  1.0000000E 00
THE F MATRIX

 1.0000000E 00        0.0
DESIRED FEEDBACK COEFFICIENTS

 2.0000000E 01        2.5320599E 01        8.0709991E 00        6.0699999E-01        8.9829999E-01        3.2139999E-01
THE GAIN =         1.4000000E 01
********************************************
THE COMPENSATOR SYSTEM MATRIX

 8.3446533E-07        1.0000000E 00
-1.2576203E 01       -4.4995995E 00
MINOR LOOP FEEDBACK COEFFICIENTS

-5.4018692E 01       -8.4977954E 00
MAJOR LOOP FEEDBACK COEFFICIENTS

 8.8752304E 01       -2.0737640E 01
 2.0606496E-04        3.2417272E-05
********************************************
```

Figure 3-38 Serial Compensator Design - Test Two

$$u(t) = [1 \quad 0]\, \underset{\sim}{v}(t) + [-54 \quad -8.5]\, \underset{\sim}{y}(t)$$

$$\underset{\sim}{y}(t) = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}\; \underset{\sim}{x}(t)$$

Step 2.5

Again it is relatively straightforward to rearrange the equations in an augmented system form suitable for simulation using the subprogram GTRESP. For completeness the result is given here.

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -54. & -23.5 & -23 & -9 & 1 & 0 \\ 88.75 & -20.74 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -12.58 & -4.5 \end{bmatrix}$$

$$\underset{\sim}{b}_1^T = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 14]$$

$$\text{gain} = 1$$

$$\underset{\sim}{c} = [20 \quad 10 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\underset{\sim}{k}^T = \underset{\sim}{0}$$

$$\underset{\sim}{x}(t_0) = \underset{\sim}{0}$$

The time specifications are chosen to be

$$t_0 = 0 \qquad t_f = 10$$

$$dt = 0.002 \qquad FREQ = 100$$

and the control and data cards for the graphical time response subprogram with a unit step input are

```
// (standard OS JOB card),TIME=2
//‸‸EXEC‸LINCONF
//FORT.SYSIN‸DD‸*
       SUBROUTINE RFIND(T,R)
       R = 1.0
       RETURN
       END
/*
//LINK.SYSIN‸DD‸*
‸‸INCLUDE‸SYSLIB(GTRESP)
/*
//GO.SYSIN‸DD‸*
SERCOM TEST TWO   06
0.0      1.0      0.0      0.0      0.0      0.0
0.0      0.0      1.0      0.0      0.0      0.0
0.0      0.0      0.0      1.0      0.0      0.0
-54.     -23.5    -23.     -9.      1.0      0.0
88.75    -20.74   0.0      0.0      0.0      1.0
```

AD-A081 770 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2

A USER'S MANUAL FOR LINEAR CONTROL PROGRAMS ON IBM/360.(U)

DEC 79 B DESJARDINS

UNCLASSIFIED NL

3 OF 3

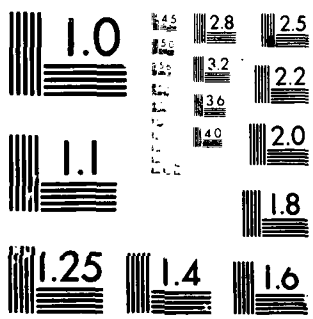
AD A081770

END

DATE FILMED

4-80

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
0.0     0.0     0.0     0.0     -12.58     -4.5
0.0     0.0     0.0     0.0     0.0        14.0
20.0    10.
0.0
1.0
0.0
0.0     10.     0.002   100.
Y
/*
```

Results in Fig. 3-39 are very similar to those obtained for the Luenberger Observer system. The response can be compared against the original specifications. If unsatisfactory, the designer can redo the problem using different pole locations. As mentioned at the end of the previous example, it might be good to find out if any mistake was made by verifying the location of the closed-loop poles. This is easily accomplished by running the subprogram STVAR for open-loop calculations for the above augmented system. The data deck consists of the problem identification, the system order, $\underset{\sim}{A}$, $\underset{\sim}{b}$ and $\underset{\sim}{C}$ matrices and two blank cards. The complete computer deck is

```
// (standard OS JOB card)
// ^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(STVAR)
/*
//GO.SYSIN ^DD^*
```

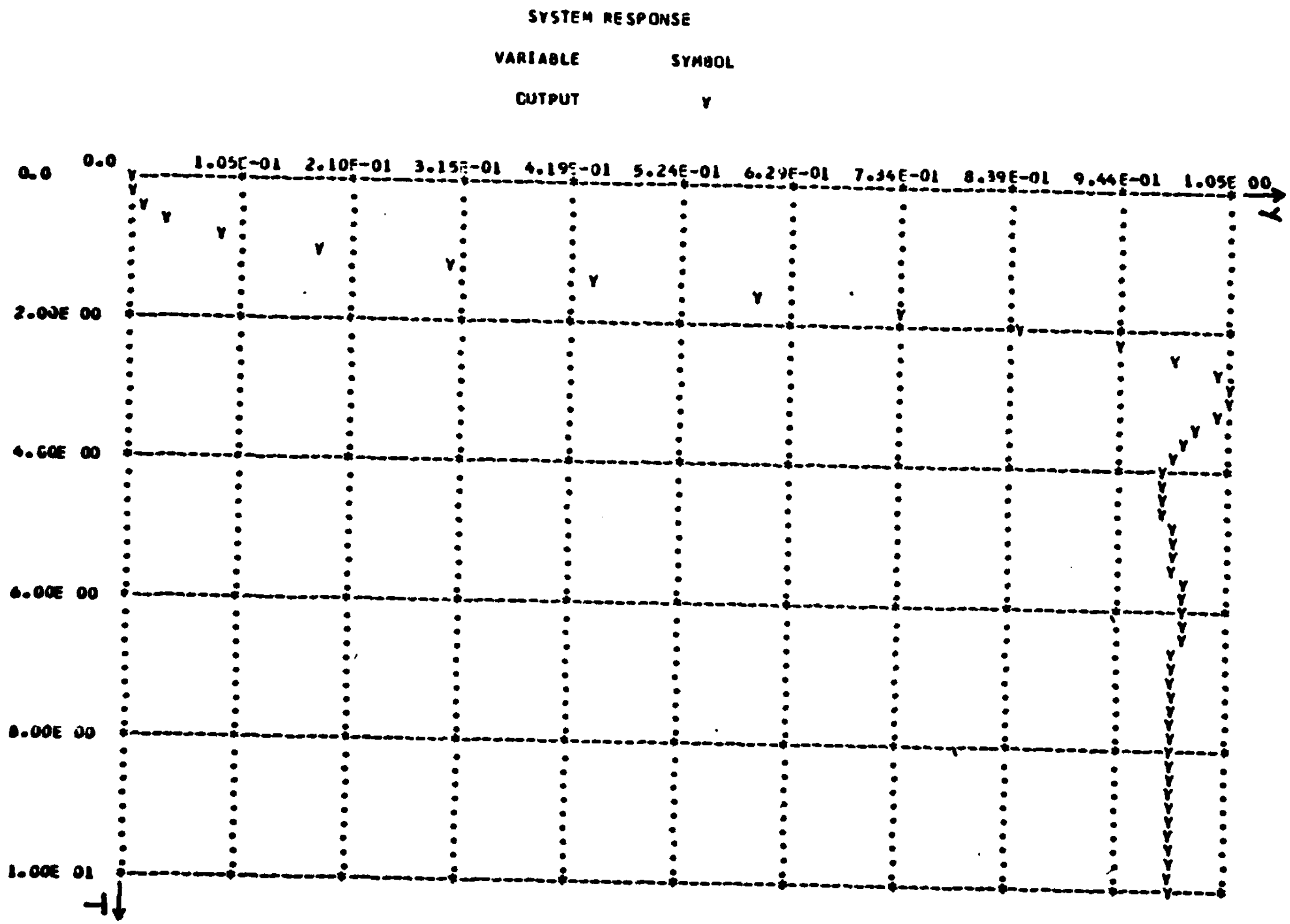


Figure 3-39 GTRESP for SERCOM Test Two

```
SERCOM TEST 2   06
0.0     1.0     0.0     0.0     0.0     0.0
0.0     0.0     1.0     0.0     0.0     0.0
0.0     0.0     0.0     1.0     0.0     0.0
-54.    -23.5   -23.    -9.0    1.0     0.0
88.75   -20.74  0.0     0.0     0.0     1.0
0.0     0.0     0.0     0.0     -12.58  -4.5
0.0     0.0     0.0     0.0     0.0     14.0
20.     10.
(blank card)
(blank card)
/*
```

Note that the gain K is carried inside the $\underset{\sim}{b}^T$ matrix as required by the equations representing the final compensated system. Results presented in Fig. 3-40 show that the roots are very close to their originally specified locations.

7. <u>Optimal Control/Kalman Filters (RICATI)</u>

RICATI is a double-precision subprogram used to solve the Riccati differential equations

$$\dot{\underset{\sim}{P}}(t) = -\underset{\sim}{P}(t)\underset{\sim}{A} - \underset{\sim}{A}^T\underset{\sim}{P}(t) + \underset{\sim}{P}(t)\,\underset{\sim}{B}\underset{\sim}{R}^{-1}\underset{\sim}{B}^T\underset{\sim}{P}(t) - \underset{\sim}{Q} \qquad (1)$$

and/or

$$\dot{\underset{\sim}{\bar{P}}}(t) = \underset{\sim}{A}\underset{\sim}{\bar{P}}(t) + \underset{\sim}{\bar{P}}\underset{\sim}{A}^T - \underset{\sim}{\bar{P}}(t)\underset{\sim}{C}^T\underset{\sim}{\bar{R}}^{-1}\underset{\sim}{C}\underset{\sim}{\bar{P}}(t) + \underset{\sim}{B}\underset{\sim}{\bar{Q}}\underset{\sim}{B}^T \qquad (2)$$

```
STATE VARIABLE FEEDBACK
PROBLEM IDENTIFICATION -      SERCOM TEST 2
***************************************************

THE A MATRIX

 0.0              1.0000000E 00    0.0              0.0              0.0              0.0
 0.0              0.0              1.0000000E 00    0.0              0.0              0.0
 0.0              0.0              0.0              1.0000000E 00    0.0              0.0
-5.4000000E 01   -2.3530000E 01   -2.3000000E 01   -9.0000000E 00    1.0000000E 00    0.0
 [illegible]     -2.0735990E 01    0.0              0.0              0.0              1.0000000E 00
 0.0              0.0              0.0              0.0             -1.2580000E 01   -4.5000000E 00

THE B MATRIX

 0.0              0.0              0.0              0.0              0.0              1.4000000E 01

PLANT IS NUMERICALLY UNCONTROLLABLE        MAX. DEVIATION =  1.42E-02     **********
***************************************************

OPEN-LOOP CALCULATIONS

DENOMINATOR COEFFICIENTS - IN ASCENDING POWERS OF S

 [illegible]      5.4320547E 02    4.6982959E 02    2.4021999E 02    7.6079987E 01    1.3500000E 01
 1.0000000E 00

THE ROOTS ARE              REAL PART          IMAGINARY PART
                        -[illegible]E 00       0.0
                        -4.0417271E 00         0.0
                        -1.0151298E 00        -2.0039644E 00
                        -1.0151298E 00         2.0039644E 00
                        -1.5692173E 00        -1.5724474E-01
                        -1.5692176E 00         1.5724474E-01

THE C MATRIX     *****

 2.0000000E 01    1.0000000E 01    0.0              0.0              0.0              0.0

NUMERATOR COEFFICIENTS - IN ASCENDING POWERS OF S

 2.8000000E 02    1.4000000E 02

THE ROOTS ARE              REAL PART          IMAGINARY PART
                        -2.0000000E 00         0.0
```

Figure 3-40 Verification of Results for SERCOM Test Two Using STVAR for Open-Loop Calculations

to obtain the gain matrix

$$G_c(t) = R^{-1}B^TP(t) \tag{3}$$

or/and

$$G_f(t) = \bar{R}^{-1}C\bar{P}(t) \tag{4}$$

Equations (1) and (3) pertain to the solution of the state-regulator problem while (2) and (4) occur in the continuous Kalman filter algorithm. For convenience a brief discussion of each subject is included. First the state-regulator problem: given a linear, time-invariant system [9]

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$

$$y(t) = C\,x(t)$$

where $u(t)$ is not constrained, a control law is to be found such that the quadratic cost function

$$J = \frac{1}{2}[x^T(t_f)P_fx(t_f)] + \frac{1}{2}\int_{t_0}^{t_f}[x^T(t)Qx(t) + u^T(t)Ru(t)]dt$$

is minimized. Such an optimal control exists, provided that $\underset{\sim}{P}_f$ and $\underset{\sim}{Q}$ are positive semidefinite and $\underset{\sim}{R}$ is positive definite, and is given by

$$\underset{\sim}{u}(t) = -\underset{\sim}{R}^{-1}\underset{\sim}{B}^T\underset{\sim}{P}(t)\ \underset{\sim}{x}(t) \triangleq -\underset{\sim}{G}_c(t)\ \underset{\sim}{x}(t)$$

where $\underset{\sim}{P}(t)$ is the unique solution of the differential Riccati equation

$$\dot{\underset{\sim}{P}}(t) = -\underset{\sim}{P}(t)\underset{\sim}{A} - \underset{\sim}{A}^T\underset{\sim}{P}(t) + \underset{\sim}{P}(t)\underset{\sim}{B}\underset{\sim}{R}^{-1}\underset{\sim}{B}^T\underset{\sim}{P}(t) - \underset{\sim}{Q}$$

with the boundary condition $\underset{\sim}{P}(t_f) = \underset{\sim}{P}_f$; t_f is a specified value. The RICATI subprogram is used to determine the control gain matrix

$$\underset{\sim}{G}_c(t) = \underset{\sim}{R}^{-1}\underset{\sim}{B}^T\underset{\sim}{P}(t)$$

such that the closed-loop system

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\ \underset{\sim}{x}(t) + \underset{\sim}{B}\ \underset{\sim}{u}(t)$$

$$\underset{\sim}{u}(t) = -\underset{\sim}{G}_c(t)\ \underset{\sim}{x}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\ \underset{\sim}{x}(t)$$

is optimal with respect to the specified performance measure. The computer can solve for either or both the transient and

the steady-state control gains.[2] Notice that the gain matrix $G_c(t)$ output by the computer does not include the negative sign of the feedback loop.

For the second type of problem, a continuous Kalman filter is to be obtained and the subprogram RICATI is used to find the optimal filter gain matrix for the design. Here again the user has a choice of getting either or both the transient and the steady-state gains.[3] The problem to be solved is to find an optimal filter for a linear, time-invariant system [10]

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{B}\,\underset{\sim}{w}(t)$$

$$\underset{\sim}{z}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t) + \underset{\sim}{v}(t)$$

where $\underset{\sim}{v}(t)$, the measurement noise, is uncorrelated and has covariance matrix $\underset{\sim}{Q}$. The random process forcing input w(t)

---

[2]The conditions sufficient for steady-state control to exist are that the system be completely controllable, i.e., the matrix $[\underset{\sim}{B}\ \ \underset{\sim}{A}\underset{\sim}{B}\ \ \ldots\ \ \underset{\sim}{A}^{n-1}\underset{\sim}{B}]$ be of rank n where n is the order of the plant, that no terminal cost be considered in the cost function and that $\underset{\sim}{A}$ and $\underset{\sim}{B}$ be time-invariant. [9]

[3]Sufficient conditions for steady-state filter gains to exist are [10]:

(a) the plant must be completely observable
(b) the plant must be time-invariant, i.e., $\underset{\sim}{A}$, $\underset{\sim}{F}$ and $\underset{\sim}{C}$ are independent of time
(c) the random processes $\underset{\sim}{v}(t)$ and $\underset{\sim}{w}(t)$ are stationary, i.e., $\underset{\sim}{R}$ and $\underset{\sim}{Q}$ are constant.

is also uncorrelated and has covariance matrix $\underset{\sim}{R}$. The expected values of the initial states are

$$\bar{\underset{\sim}{x}}_o = E[\underset{\sim}{x}(t_o)]$$

The solution is obtained by choosing the filter gain matrix

$$\underset{\sim}{G}_f(t) = \underset{\sim}{\bar{R}}^{-1}\underset{\sim}{C}\underset{\sim}{\bar{P}}(t)$$

such that the plant, measurement and Kalman filter are

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{B}\,\underset{\sim}{w}(t)$$

$$\underset{\sim}{z}(t) = C\,\underset{\sim}{x}(t) + \underset{\sim}{v}(t)$$

$$\dot{\hat{\underset{\sim}{x}}}(t) = \underset{\sim}{A}\,\underset{\sim}{\hat{x}}(t) + \underset{\sim}{G}_f(t)[\underset{\sim}{z}(t) - \underset{\sim}{C}\,\underset{\sim}{\hat{x}}(t)]$$

These equations are also presented in block diagram form in Figure 3-41.

The purpose of the subprogram RICATI is to solve the differential Riccati equation

$$\dot{\bar{\underset{\sim}{P}}}(t) = \underset{\sim}{A}\,\bar{\underset{\sim}{P}}(t) + \bar{\underset{\sim}{P}}(t)\underset{\sim}{A}^T + \underset{\sim}{B}\bar{\underset{\sim}{Q}}\underset{\sim}{B}^T - \bar{\underset{\sim}{P}}(t)\underset{\sim}{C}^T\underset{\sim}{R}^{-1}\underset{\sim}{C}\bar{\underset{\sim}{P}}(t)$$

with initial condition

$$\underset{\sim}{\overline{P}}(t_o) = \underset{\sim}{\overline{P}}_o = E[(\underset{\sim}{\hat{x}}(t_o) - \underset{\sim}{\overline{x}}_o) \cdot (\underset{\sim}{\hat{x}}(t_o) - \underset{\sim}{\overline{x}}_o)^T]$$

to calculate the filter gain matrix $\underset{\sim}{G}_f(t)$.

a. Input

A common input format applies to both state-regulator and Kalman filter problems. However the matrix definitions differ.

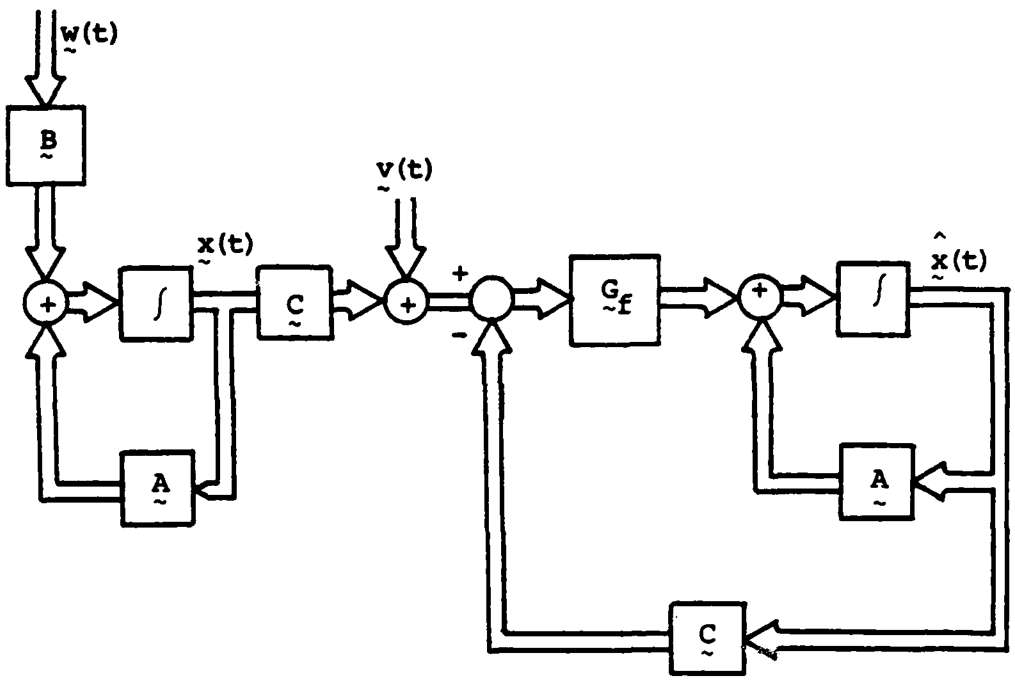


Fig 3-41 Continuous Kalman Filter Block Diagram

(1) Basic Input

The input data deck first card contains the problem identification, the order of the plant ($N \leq 10$), the number of control inputs ($M \leq 10$) and the number of measured

outputs ($L \leq 10$). Since these numbers define the dimensions of each subsequent matrix, extra care is suggested. Next the plant matrix $\underset{\sim}{A}$ ($N \times N$) is entered one row at a time. Similarly the control matrix $\underset{\sim}{B}^T$ ($M \times N$) and the observable output matrix $\underset{\sim}{C}$ ($L \times N$) are given. The above forms the basic input and needs only be included once.

(2) Control Option Input

This portion of the data is used when solving state-regulator problems. The letter C is printed in the first column of the first card to indicate that option control is selected. On this same card, if and only if transient gains are desired, the user gives the initial time t_0, the final time t_f and the number of time points of the control gain matrix (NPOINT) to be printed. If the steady-state solution only is desired, the letter C still appears in column one but the rest of the card is left blank.

Next the control weighting matrix $\underset{\sim}{R}$ ($M \times M$) is entered, followed by the state weighting matrix $\underset{\sim}{Q}$ ($N \times N$). If and only if the transient response of the gains was requested, by assigning non-zero values to t_0, t_f and n points, the terminal boundary condition matrix $\underset{\sim}{P}(t_f)$ ($N \times N$) is given last.

(3) Filter Option Input

The first card of this portion of the data deck indicates a Kalman filter problem by the letter F punched in column one. As for the control option input, the time interval and number of points of the filter gain

matrix transient response to be output are also entered on that first card, if and only if the transient response is desired. Next, the measurement noise covariance matrix $\underset{\sim}{R}$ $(L \times L)$ and the random input covariance matrix Q $(M \times M)$ are entered, one row at a time. Finally, if and only if the transient filter gain solution was requested by assigning non-zero values to t_0, t_f and NPOINT the initial boundary condition matrix $\underset{\sim}{\overline{P}}(t_0)$ $(N \times N)$ is given.

(4) Problem Termination Card

The user may ask for several different computer solutions of the same basic problem by stacking the control input cards for transient response and the control input cards for steady-state solution, or the filter input cards for steady-state solution and the filter input cards for transient response. Termination of a given problem is indicated by a blank card. As usual, many problems can be executed under the same run by placing the complete data decks one on top the other.

The following input format table summarizes the above.

Entry	Input Description	Format	Columns Used
1 Basic	Problem identification,	5A4,	1-20,
	order of the plant $(N \leq 10)$	I2,	21-22,
	number of control inputs $(M \leq 10)$,	I2,	23-24,
	number of measurements $(L \leq 10)$.	I2	25-26

Entry	Input Description	Format	Columns Used
2 Basic	Plant matrix $\underset{\sim}{A}$ $(N \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
3 Basic	Distribution matrix $\underset{\sim}{B}^T$ $(M \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
4 Basic	Measurement matrix $\underset{\sim}{C}$ $(L \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
5 Control Option	Letter C, initial time t_0, final time t_f, number of points (NPOINT)	A1, F10.3, F10.3 I3	1, 11-20, 21-30, 31-32-33
6 Control Option	Control weighting matrix $\underset{\sim}{R}$ $(M \times M)$ (one row per card if $M \leq 8$; one row per two cards for $M > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
7 Control Option	State weighting matrix $\underset{\sim}{Q}$ $(N \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
8 Iff NPOINT $\neq 0$ Control Option	Terminal boundary matrix $\underset{\sim}{P}(t_f)$ $(N \times N)$ (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
9 Filter Option	Letter F, initial time t_0, final time t_f, number of points NPOINT	A1, F10.3, F10.3, I3	1, 11-20, 21-30, 31-32-33
10 Filter Option	Measurement noise covariance matrix $\bar{\underset{\sim}{R}}$ $(L \times L)$ (one row per card for $L \leq 8$; one row per two cards for $L > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
11 Filter Option	Random input covariance matrix $\bar{\underset{\sim}{Q}}$ $(M \times M)$ (one row per card for $M \leq 8$; one row per two cards for $M > 8$)	8E10.0	1-10, 11-20, 21-30, etc.

Entry	Input Description	Format	Columns Used
12 Iff NPOINT ≠0 Filter Option	Initial boundary value matrix $\underset{\sim}{P}(t_0)$ (N × N) (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
13	(blank card) (indicates problem termination)	8E10.0	(blank)

Table XVI - Input Format Table for RICATI

b. Output

The problem identification and the $\underset{\sim}{A}$, $\underset{\sim}{B}^T$ and $\underset{\sim}{C}$ matrices are listed for reference. Then the option requested is indicated and the $\underset{\sim}{R}$, $\underset{\sim}{Q}$ and $\underset{\sim}{P}$ matrices are printed. Finally, "steady-state solution" or "transient response" is printed, followed by the gain matrix $\underset{\sim}{G}_f$ or $\underset{\sim}{G}_c$.

c. Examples

Two problems are worked out to illustrate the use of this subprogram.

(1) Example One

In the first case we assume the plant

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = u(t)$$

and wish to determine what must the control gains be to minimize the performance measure

$$J = \int_0^{\infty} [q_{11}x_1^2(t) + q_{22}x_2^2(t) + Ru^2(t)]\, dt$$

where the weighting factors are $q_{11} = 4.0$, $q_{22} = 0$ and $R = 50$. The control option is used. The elements necessary for the data deck are:

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad \underset{\sim}{B}^T = [0 \quad 1]$$

$$\underset{\sim}{C} = [0 \quad 0]$$

(Note that $\underset{\sim}{C}$ is not used in the calculations but must be included since the input table requires it.)

$$R = 50.$$

$$\underset{\sim}{Q} = \begin{bmatrix} 4.0 & 0 \\ 0 & 0 \end{bmatrix}$$

Both the steady-state and transient solution are desired. For the transient part of the problem, $t_0 = 0.0$, $t_f = 10.0$ and NPOINT = 020 are selected with the initial condition $\underset{\sim}{P}(t_f) = 0$.

The control and data cards are then

// (standard OS JOB card)

// ^EXEC^LINCON

//LINK.SYSIN^DD^*

```
^^INCLUDE^SYSLIB(RICATI)
/*
//GO.SYSIN^DD^*
RICATI CONTROL TEST  020101
0.0    1.0
0.0    0.0
0.0    1.0
1.0    0.0
C
50.
4.0    0.0
0.0    0.0
C      0.0            10.0      020
50.
4.0    0.0
0.0    0.0
0.0    0.0
0.0    0.0
(blank card)
/*
```

The solution in Fig. 3-42 shows the requested steady-state and transient response.

(2) Example Two

The second problem is to find the optimal Kalman filter gain matrix for the following system:

$$\dot{\underset{\sim}{x}}(t) = \begin{matrix} 0 & 1 \\ 0 & -4.6 \end{matrix} \; \underset{\sim}{x}(t) + \begin{matrix} 0 \\ 0.1 \end{matrix} \; w(t)$$

```
OPTIMAL CONTROL/KALMAN FILTER PROGRAM
PROBLEM IDENTIFICATION - RICATI CONTROL TEST
***********************************************
  THE A MATRIX
0.0                 1.000000000 00
0.0                 0.0
  THE B MATRIX
0.0                 1.000000000 00
  THE C MATRIX
1.000000000D 00     0.0
***********************************************
*** CONTROL OPTION ***

  THE R MATRIX
5.000000000 01
  THE Q MATRIX
4.000000000 00      0.0
0.0                 0.0
STEADY STATE SOLUTION
  GAINS
2.828426140-01      7.521196010-01
***********************************************
*** CONTROL OPTION ***

  THE R MATRIX
5.000000000D 01
  THE Q MATRIX
4.000000000 00      0.0
0.0                 0.0
  INITIAL CONDITIONS
0.0                 0.0
0.0                 0.0
***********************************************
  TIME =.       1.20000000D 01
GAINS
0.0                 0.0
  TIME =.       1.14000001D 01
GAINS
1.403242550-02      5.472364400-03
  TIME =.       1.08000063D C1
GAINS
5.623951600-02      4.438778840-02
  TIME =.       1.02000004D 01
GAINS
1.214259300-01      1.437892900-01
  TIME =.       9.60000050D C0
GAINS
1.934566330-01      3.028165130-01
  TIME =.       9.00000063D C0
GAINS
2.492548240-01      4.775958020-01
  TIME =.       8.40000075D 0C
GAINS
2.773072950-01      6.136688570-01
  TIME =.       7.80000088D 00
GAINS
2.831003900-01      6.919276840-01
  TIME =.       7.20000100D C0
GAINS
2.798658030-01      7.261927910-01
  TIME =.       6.60000113D C0
GAINS
2.755197080-01      7.370992770-01
  TIME =.       6.00000125D 0C
GAINS
2.732021230-01      7.389650480-01
  TIME =.       5.40000138D 00
GAINS
2.731831320-01      7.389602520-01
  TIME =.       4.80000150D C0
GAINS
2.746721080-01      7.396905890-01
  TIME =.       4.20000163D 00
GAINS
2.767589630-01      7.415017080-01
  TIME =.       3.60000175D 00
GAINS
2.787841950-01      7.439051700-01
  TIME =.       3.00000188D C0
GAINS
2.804067100-01      7.463329320-01
  TIME =.       2.40000200D 0C
GAINS
2.815364860-01      7.483850950-01
  TIME =.       1.80000213D 00
GAINS
2.822307670-01      7.499113800-01
  TIME =.       1.20000225D C0
GAINS
2.826049710-01      7.509277650-01
  TIME =.       6.00002378D-C1
GAINS
2.827759260-01      7.515369650-01
  TIME =.       2.50339508D-06
GAINS
2.828353740-01      7.518635080-01
***********************************************
```

Figure 3-42 Control Option Test for RICATI

The random input w(t) is white noise with variance Q = 10. The observed variable is given by

$$z(t) = [1 \quad 0]\, \underset{\sim}{x}(t) + v(t)$$

where v(t) is also white noise with variance $R = 10^{-7}$. From the above it is easy to extract the data necessary to solve the problem by the use of RICATI. Writing down the elements one gets:

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 \\ 0 & -4.6 \end{bmatrix} \qquad \underset{\sim}{B}^T = [0 \quad .1]$$

$$\underset{\sim}{C} = [1 \quad 0]$$

$$R = 10^{-7}$$

$$Q = 10$$

The initial condition matrix $P(t_0)$ is chosen to be

$$\underset{\sim}{\overline{P}}(t_0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The time is specified as being $t_0 = 0.0$ and $t_f = 0.5$ and a number of points to be output is NPOINT = 10. The computer cards to solve both for the transient and steady-state are

```
// (standard OS JOB card)
// ^ EXEC^LINCON
//LINK.SYSIN^DD^*
 ^ ^ INCLUDE^SYSLIB(RICATI)
/*
//GO.SYSIN^DD^*
RICATI FILTER TEST  020101
0.0       1.0
0.0       -4.6
0.0       0.1
1.0       0.0
F         0.0       0.5       010
0.0000001
10.
F
0.0000001
10.
/*
```

Results presented in Fig. 3-43 indicate that the algorithm used by the computer to find the steady-state gains is not good enough for the problem. The transient response final values are used as steady state gains.

8. Discrete Time Kalman Filter (Kalman)

This double-precision subprogram is used to calculate the discrete Kalman filter gain matrix $\underset{\sim}{G}_k$. The theory of the discrete Kalman filter can be obtained from many textbooks and articles and is not reproduced here. For example see [10] and [11].

```
     OPTIMAL CONTROL/KALMAN FILTER PROGRAM
     PROBLEM IDENTIFICATION - RICATI FILTER TEST
*********************************************
      THE A MATRIX
     0.0                  1.000000000D 00
     0.0                 -4.600000000D 00
      THE B MATRIX
     0.0                  1.000000000D-01
      THE C MATRIX
     1.000000000D 00      0.0
*********************************************
     *** FILTER OPTION ***

      THE R MATRIX
     1.000000000D-07
      THE Q MATRIX
     1.000000000D 01
      INITIAL CONDITIONS
     0.0                  0.0
     0.0                  0.0
*********************************************
      TIME =.       0.0
     GAINS
     0.0                  0.0
      TIME =.       4.999998960D-02
     GAINS
     1.717204510 01       6.858274780D 02
      TIME =.       9.999997910D-02
     GAINS
     4.036759960D 01      7.822256930D 02
      TIME =.       1.499999690D-01
     GAINS
     3.990694580D 01      8.080184440D 02
      TIME =.       1.999999580D-01
     GAINS
     4.037878530D 01      8.148348770D 02
      TIME =.       2.499999480D-01
     GAINS
     4.035261820D 01      8.142348580D 02
      TIME =.       2.999999370D-01
     GAINS
     4.035756700D 01      8.143682460D 02
      TIME =.       3.499999270D-01
     GAINS
     4.035728330D 01      8.143547270D 02
      TIME =.       3.999999170D-01
     GAINS
     4.035731320D 01      8.143565030D 02
      TIME =.       4.499999060D-01
     GAINS
     4.035731320D 01      8.143563460D 02
      TIME =.       4.999998960D-01
     GAINS
     4.035731300D 01      8.143563600D 02
*********************************************
     *** FILTER OPTION ***

      THE R MATRIX
     1.000000000D-07
      THE Q MATRIX
     1.000000000D 01
     UNABLE TO FIND STEADY-STATE GAINS
    PLEASE USE TRANSIENT RESPONSE OPTION
```

Figure 3-43 Filter Option Test for RICATI

The following block diagram, definitions and equations are nonetheless included to summarize the ideas and clarify the notation adopted in this discussion.

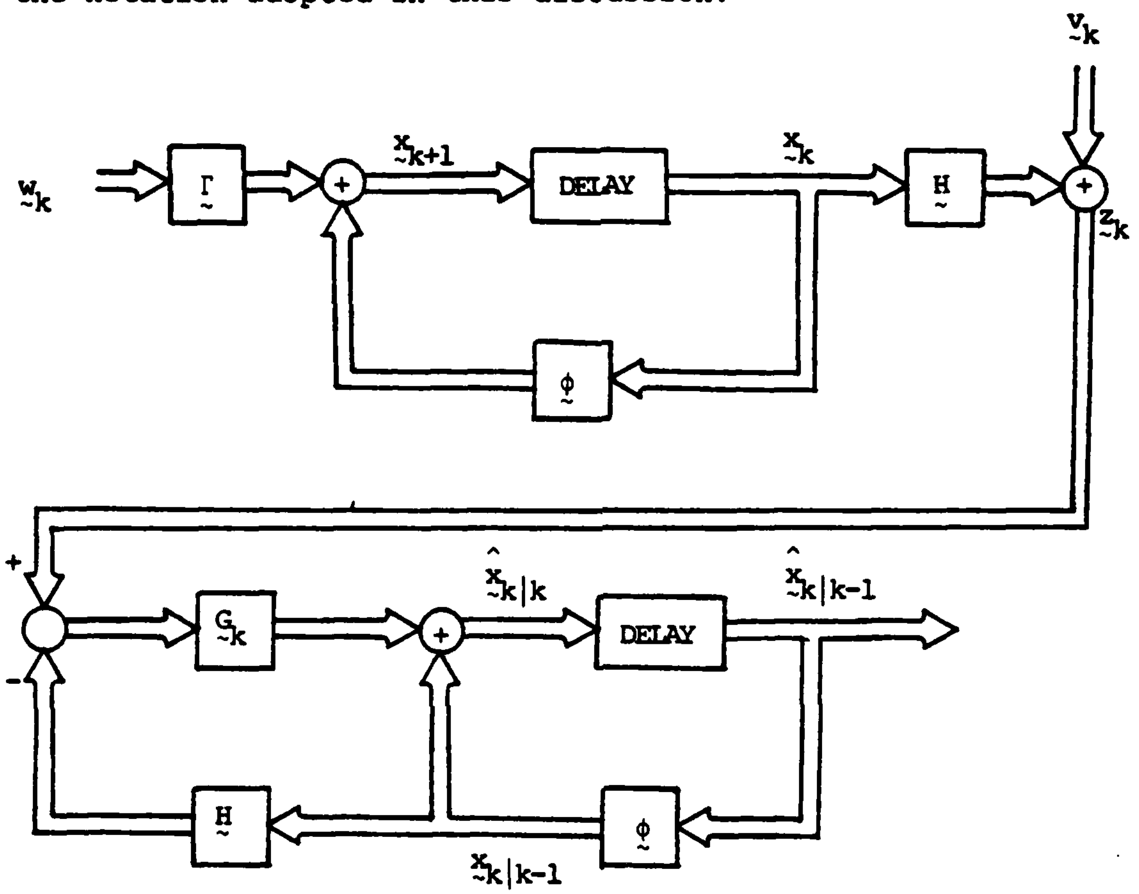


Fig 3-44 Discrete Kalman Filter Block Diagram

From the diagram, one gets the discrete time system state equation

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{\phi}\,\underset{\sim}{x}_k + \underset{\sim}{\Gamma}\,\underset{\sim}{w}_k$$

and the measurement equation

$$\underset{\sim}{z}_k = \underset{\sim}{H}\,\underset{\sim}{x}_k + \underset{\sim}{v}_k$$

Each element can be briefly defined and the matrix dimensions noted as:

$\underset{\sim}{x}_k$: state vector $(N \times 1)$

$\underset{\sim}{\phi}$: transition matrix $(N \times N)$

$\underset{\sim}{w}_k$: system random input $(L \times 1)$

$\underset{\sim}{\Gamma}$: distribution matrix $(N \times L)$

$\underset{\sim}{z}_k$: measurement vector $(M \times 1)$

$\underset{\sim}{H}$: observation matrix $(M \times N)$

$\underset{\sim}{v}_k$: measurement noise $(M \times 1)$

$\underset{\sim}{G}_k$: gain matrix $(N \times M)$

The problem is to minimize

$$J_1 = E[(\underset{\sim}{x}_k - \hat{\underset{\sim}{x}}_{k|k})^T (\underset{\sim}{x}_k - \hat{\underset{\sim}{x}}_{k|k})]$$

with respect to $\underset{\sim}{G}_k$. Note that J_1 is a scalar.[4]

---

[4] J_1 actually is the trace of the cost function

$$\underset{\sim}{J} = E[(\underset{\sim}{x}_k - \hat{\underset{\sim}{x}}_{k|k})(\underset{\sim}{x}_k - \hat{\underset{\sim}{x}}_{k|k})^T]$$

where $\underset{\sim}{J}$ is a $(N \times N)$ matrix.

The solution to the problem is

$$\underset{\sim}{G}_k = \underset{\sim}{P}_{k|k-1} \underset{\sim}{H}^T [\underset{\sim}{H} P_{k|k-1} \underset{\sim}{H}^T + \underset{\sim}{R}]^{-1} \tag{1}$$

$$\underset{\sim}{P}_{k|k} = [\underset{\sim}{I} - \underset{\sim}{G}_k \underset{\sim}{H}]\, \underset{\sim}{P}_{k|k-1} \tag{2}$$

$$\underset{\sim}{P}_{k+1|k} = \underset{\sim}{\phi}\, \underset{\sim}{P}_{k|k}\, \phi^T + \underset{\sim}{Q} \tag{3}$$

where

$$\underset{\sim}{x}_{k|k} = \hat{\underset{\sim}{x}}_{k|k-1} + \underset{\sim}{G}_k [\underset{\sim}{z}_k - \underset{\sim}{H} \hat{\underset{\sim}{x}}_{k|k-1}] \tag{4}$$

and

$$\hat{\underset{\sim}{x}}_{k|k-1} = \underset{\sim}{\phi}\, \hat{\underset{\sim}{x}}_{k-1|k-1} \tag{5}$$

given the initial conditions

$$\hat{\underset{\sim}{x}}_{0|-1} = E[\underset{\sim}{x}(0)]$$

and

$$\underset{\sim}{P}_{0|-1} = E[(\underset{\sim}{x} - \underset{\sim}{x}_{0|-1})^2]$$

The terms associated with the above equations are defined as

$\underset{\sim}{P}_{k|k}$: (N × N) matrix of the covariance of error of the estimate at k given observations at times up to and including time k.

$\underset{\sim}{P}_{k|k-1}$: $(N \times N)$ matrix of the covariance of error of the one-step prediction at k given observations at times up to and including time (k-1)

$\underset{\sim}{R}$: $(M \times N)$ covariance matrix of the measurement noise

$\underset{\sim}{Q}$: $(N \times N)$ covariance matrix of the random input

The matrix

$$\underset{\sim}{Q} = \underset{\sim}{\Gamma} \, E(\underset{\sim}{W}_k \underset{\sim}{W}_k^T) \, \underset{\sim}{\Gamma}^T$$

is computed from the parameters $\underset{\sim}{\Gamma}^T$ $(L \times N)$ and $E[\underset{\sim}{W}_k \underset{\sim}{W}_k^T]$ $(L \times L)$.

The purpose of the subprogram KALMAN is to solve the recurrence relations (1), (2) and (3) for a specified number of iterations N and print the filter gain matrix G_k as a function of k.

a. Input

Since many problems are encountered where the designer must compensate for time-varying environment by letting the covariance of the observation noise be variable, it was decided to permit the user to define the $\underset{\sim}{R}$ $(M \times M)$ matrix with an external subroutine. The subprogram KALMAN thus is accessible under Mode Two of operation only. Also note that the subprogram is double-precision.

The first input to be entered is the covariance of the observation noise via the double-precision subroutine RDEF(R,NP,M) performed by the main program where M is the order of the matrix. The parameters NP and M are directly

available from the main program and must not be assigned any value here, i.e., leave them as NP and M. The subroutine is then

```
SUBROUTINE RDEF(R,NP,M)
IMBLICIT REAL*8  (A-H, O-Z)
DIMENSION R(20,20)
      FORTRAN statements defining
      the R matrix (see example part c.)
RETURN
END
```

Next the data deck is punched. The problem identification, the order of the system N, the dimension of the random input vector L and the number of outputs M are given on the first card and are followed by the $\underset{\sim}{\phi}(N \times N)$ matrix, the $\underset{\sim}{\Gamma}^T$ $(L \times N)$ matrix, the $E[\underset{\sim}{W}\underset{\sim}{W}^T]$ $(L \times L)$ matrix, the $\underset{\sim}{H}$ $(M \times N)$ matrix and the initial condition matrix $\underset{\sim}{P}_{0|-1}$ $(N \times N)$ in accordance with the input format table shown below.

Entry	Input Description	Format	Columns Used
1	Problem identification, order of the system ($N \leq 10$), dimension of the random input vector ($L \leq 10$), number of measurements ($M \leq 10$)	5A4, 3I2	1-20, 21-22, 23-24, 25-26
2	$\underset{\sim}{\phi}$ $(N \times N)$ matrix (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
3	$\underset{\sim}{\Gamma}^T$ $(L \times N)$ matrix (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.

Entry	Input Description	Format	Columns Used
4	$E[\underset{\sim}{W}\underset{\sim}{W}^T]$ $(L \times L)$ matrix (one row per card for $L \le 8$; one row per two cards for $L > 8$)	8E10.0	1-10, 11-20 21-30, etc.
5	$\underset{\sim}{H}$ $(M \times N)$ matrix (one row per card for $N \le 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
6	Number of time points to be performed (NP)	8E10.0	1-10
7	$\underset{\sim}{P}_{0\|-1}$ $(N \times N)$ matrix (one row per card for $N \le 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.

Table XVII - Input Format Table for KALMAN

b. Output

The problem identification, the discrete system $\underset{\sim}{\phi}$ matrix, the transpose of the gamma matrix, the $E[\underset{\sim}{W}\underset{\sim}{W}^T]$ matrix (listed as the W matrix on the printout), the measurement matrix $\underset{\sim}{H}$, the <u>initial value</u> of the observation noise covariance matrix (at NP = 0) and the initial condition matrix are listed for reference. Then the filter gain matrix is printed as a function of the time index k, from $k = 0$ to $k = NP$.

c. Example

It is desired to estimate position and velocity from noisy position measurements only. The system equations are

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & .5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} .125 \\ .5 \end{bmatrix} w(k)$$

$$z(k) = [1 \quad 0] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + v(k)$$

where the perturbation acceleration w(k) has a root mean-square magnitude of 2 meters per second.

From the above information, one can see that

$$\underset{\sim}{\phi} = \begin{bmatrix} 1 & .5 \\ 0 & 1 \end{bmatrix}$$

$$\underset{\sim}{\Gamma}^T = [.125 \quad .5]$$

$$E[\underset{\sim}{W}\underset{\sim}{W}^T] = \text{mean-square magnitude of the perturbation acceleration}$$

$$= 4.$$

The matrix $\underset{\sim}{P}_{0|-1}$ is assumed to be

$$\underset{\sim}{P}_{0|-1} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

and the covariance of the observation noise is assumed to be

$$R = \begin{cases} 4 + (-\frac{1}{2})^{(NP)} & \text{for } 0 \leq NP \leq 10 \\ 4 & \text{for } 10 < NP \end{cases}$$

The number of time points to be computed is chosen to be 20 and the following computer card deck is set up:

```
// (standard OS JOB card)
//^EXEC^LINCONF
//FORT.SYSIN^DD^*
       SUBROUTINE RDEF(R,NP,M)
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION R(20,20)
       DO  1  I=1,M
       DO  1  J=1,M
       IF(NP.LE.10)    R(I,J)=4.+(-0.5)**NP
    1  IF(NP.GT.10)  R(I,J)=4.
       RETURN
       END
/*
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(KALMAN)
^^ENTRY^KALMAN
/*
//GO.SYSIN^DD^*
KALMAN TEST        020101
1.0      0.5
0.0      1.0
0.125    0.5
4.0
1.0      0.0
10.0     0.0
```

```
  DISCRETE TIME KALMAN FILTER PROGRAM
 PROBLEM IDENTIFICATION = KALMAN TEST
*********************************************

  THE PHI MATRIX
 1.00000000D 00      5.00000000D-01
 0.0                 1.00000000D 00
  THE GAMMA MATRIX
 1.25000000D-01      5.00000000D-01
  THE W MATRIX
 4.00000000D 00
  THE H MATRIX
 1.00000000D 00      0.0
  THE R MATRIX
 5.00000000D 00
  INITIAL CONDITIONS
 1.00000000D 01      0.0
 0.0                 1.00000000D 01
*********************************************

  K =   0
 GAINS
 6.66666667D-01      0.0
  K =   1
 GAINS
 6.27494457D-01      5.58758315D-01
  K =   2
 GAINS
 5.94504899D-01      5.95256558D-01
  K =   3
 GAINS
 6.24958240D-01      5.28080571D-01
  K =   4
 GAINS
 5.70600281D-01      4.26178571D-01
  K =   5
 GAINS
 5.47407715D-01      3.82929507D-01
  K =   6
 GAINS
 5.21220326D-01      3.58088981D-01
  K =   7
 GAINS
 5.12172420D-01      3.52391247D-01
  K =   8
 GAINS
 5.06408851D-01      3.50671542D-01
  K =   9
 GAINS
 5.05603350D-01      3.51571689D-01
  K =  10
 GAINS
 5.05143227D-01      3.51860646D-01
  K =  11
 GAINS
 5.05300933D-01      3.52043799D-01
  K =  12
 GAINS
 5.05303983D-01      3.51961838D-01
  K =  13
 GAINS
 5.05255903D-01      3.51852161D-01
  K =  14
 GAINS
 5.05203928D-01      3.51777892D-01
  K =  15
 GAINS
 5.05166997D-01      3.51743062D-01
  K =  16
 GAINS
 5.05147578D-01      3.51732164D-01
  K =  17
 GAINS
 5.05139856D-01      3.51731055D-01
  K =  18
 GAINS
 5.05137730D-01      3.51732167D-01
  K =  19
 GAINS
 5.05137488D-01      3.51732915D-01
  K =  20
 GAINS
 5.05137561D-01      3.51733025D-01
```

Figure 3-45 Discrete Time Kalman Filter Test

```
0.0    10.0
/*
```

The results of this run are shown in Fig. 3-45.

9. Discrete Time Linear State Regulator (STREG)

This double-precision subprogram is used to compute the discrete linear regulator feedback gains $\underset{\sim}{F}(NS - K)$. The discrete linear regulator problem can be stated as [12]: given a time-invariant discrete system represented by

$$\underset{\sim}{x}(k+1) = \underset{\sim}{A}\underset{\sim}{x}(k) + \underset{\sim}{B}\underset{\sim}{w}(k)$$

where the states and controls are unconstrained, find an optimal control $\underset{\sim}{u}^*(\underset{\sim}{x}(k), k)$ that minimizes the performance index

$$\underset{\sim}{J} = \frac{1}{2}\underset{\sim}{x}^T(NS)\,\underset{\sim}{H}\,\underset{\sim}{x}(NS) + \frac{1}{2}\sum_{k=0}^{N-1}[\underset{\sim}{x}^T(k)\,\underset{\sim}{Q}\,\underset{\sim}{x}(k) + \underset{\sim}{w}^T(k)\,\underset{\sim}{R}\,\underset{\sim}{w}(k)]$$

where

$\underset{\sim}{x}(k)$: state vector $(N \times 1)$

$\underset{\sim}{A}$: coefficient matrix $(N \times N)$

$\underset{\sim}{B}$: distribution matrix $(N \times M)$

$\underset{\sim}{w}(k)$: system input vector $(M \times 1)$

$\underset{\sim}{J}$: performance index (scalar)

$\underset{\sim}{H}$: real symmetric positive semi-definite matrix ($N \times N$)

$\underset{\sim}{Q}$: real symmetric positive semi-definite matrix ($N \times N$)

$\underset{\sim}{R}$: real symmetric positive definite matrix ($M \times M$)

NS : fixed integer greater than 0 (number of stages)

After solving the problem, one realizes that the optimal feedback gains can be evaluated by solving the following two equations only:

$$\underset{\sim}{F}(NS-K) = -[\underset{\sim}{R} + \underset{\sim}{B}^T \underset{\sim}{P}(K-1)\underset{\sim}{B}]^{-1} \times [\underset{\sim}{B}^T \underset{\sim}{P}(K-1)\underset{\sim}{A}] \qquad (1)$$

$$\underset{\sim}{P}(K) = [\underset{\sim}{A} + \underset{\sim}{B}\underset{\sim}{F}(NS-K)]^T \underset{\sim}{P}(K-1)[\underset{\sim}{A} + \underset{\sim}{B}\underset{\sim}{F}(NS-K)] \qquad (2)$$

$$+ \underset{\sim}{F}^T(NS-K)\underset{\sim}{R}\underset{\sim}{F}(NS-K) + \underset{\sim}{Q}$$

where $\underset{\sim}{F}(NS-K)$ is the feedback gain matrix and $\underset{\sim}{P}(0) = \underset{\sim}{H}$.

The STREG subprogram determines the $\underset{\sim}{F}(NS-K)$ matrix for $0 < NS \leq 999$ as K varies from one to NS. It also gives the final value of the real symmetric $\underset{\sim}{P}(K)$ matrix, i.e., $\underset{\sim}{P}(NS)$. From these results the user can design the optimal discrete system

$$\underset{\sim}{x}(k+1) = \underset{\sim}{A}\ \underset{\sim}{x}(k) + \underset{\sim}{B}\ \underset{\sim}{w}(k)$$

$$\underset{\sim}{w}(k) = \underset{\sim}{F}(k)\ \underset{\sim}{x}(k)$$

where k = NS - K (a block diagram representation of the system is shown in fig. 3-46). Note that $\underset{\sim}{P}(NS)$ is presented so one can also calculate the minimum cost for the NS-stage process given some initial state x_o using the relation [17]

$$J^*_{o,N}(\underset{\sim}{x}_o) = \frac{1}{2}\, \underset{\sim}{x}_o^T\, \underset{\sim}{P}(NS)\, \underset{\sim}{x}_o$$

a. Input

The first input data card consists of the problem identification, the A matrix dimension ($N \leq 10$) and the number of inputs ($M \leq 10$). Then the $\underset{\sim}{A}$ ($N \times N$), $\underset{\sim}{B}^T$ ($M \times N$), $\underset{\sim}{H}$ ($N \times N$), $\underset{\sim}{Q}$ ($N \times N$) and $\underset{\sim}{R}$ ($M \times M$) matrices are presented one row at a time. Finally the number of stages ($0 < NS \leq 999$) is given. The following input format table further describes the required data cards.

Entry	Input Description	Format	Columns Used
1	Problem identification, system order ($N \leq 10$), number of inputs ($M \geq 10$)	5A4, 2I2	1-20, 21-22, 23-24
2	$\underset{\sim}{A}$($N \times N$) matrix (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
3	$\underset{\sim}{B}^T$($M \times N$) matrix (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
4	$\underset{\sim}{H}$ ($N \times N$) matrix (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.
5	$\underset{\sim}{Q}$ ($N \times N$) matrix (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8E10.0	1-10, 11-20, 21-30, etc.

$k \triangleq NS - K$

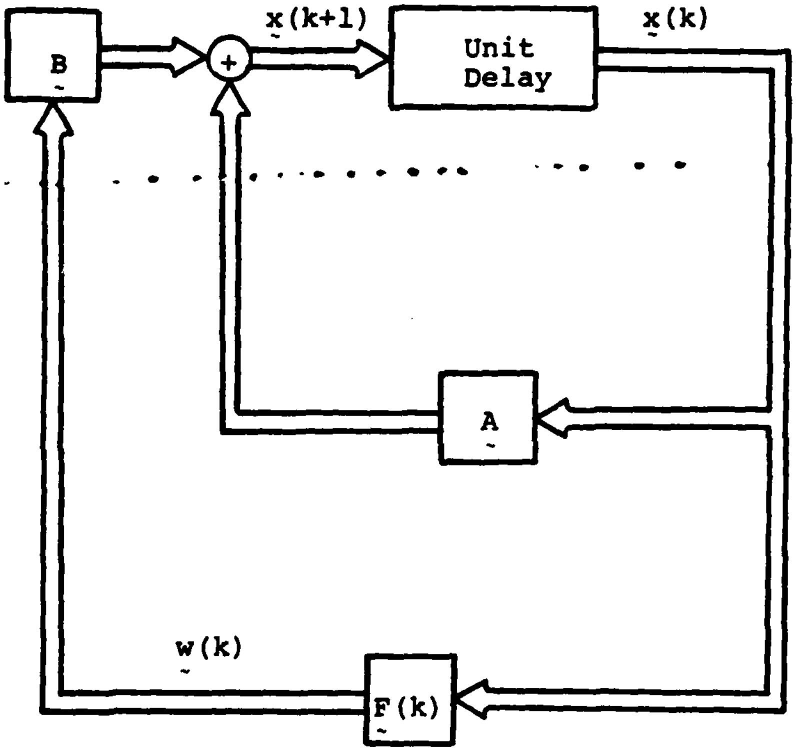


Figure 3-46 Discrete Linear Regulator Block Diagram

Entry	Input Description	Format	Columns Used
6	$\underset{\sim}{R}$ (M × M) matrix (one row per card for M $\leq$ 8; one row per two cards for M > 8)	8E10.0	1-10, 11-20, 21-30, etc.
7	Number of stages for the process (0 < NS $\leq$ 999)	I3	1-3

b. Output

The problem identification, the discrete system $\underset{\sim}{A}$ matrix, the transpose of the distribution matrix and the $\underset{\sim}{H}$, $\underset{\sim}{Q}$ and $\underset{\sim}{R}$ matrices are listed for reference. Then the feedback gain matrix $\underset{\sim}{F}$(NS - K) is printed as a function of the backward time index (NS - K) for K = 1 to K = NS. Finally, the real symmetric $\underset{\sim}{P}$(NS) matrix is given.

c. Example

Given the linear discrete system [12]

$$\underset{\sim}{x}(k+1) = \begin{bmatrix} 0.9974 & 0.0539 \\ -.1078 & 1.1591 \end{bmatrix} \underset{\sim}{x}(k) + \begin{bmatrix} 0.0013 \\ 0.0539 \end{bmatrix} w(k)$$

the feedback gain matrix $\underset{\sim}{F}$(NS - K) is to be determined which minimizes the performance measure

$$J = \frac{1}{2} \sum_{k=0}^{N-1} [0.25\ x_1^{\ 2}(k) + 0.05\ x_2^{\ 2}(k) + 0.05\ w^2(k)]$$

The data are then:

$$\underset{\sim}{A} = \begin{bmatrix} 0.9974 & 0.0539 \\ -.1078 & 1.1591 \end{bmatrix}$$

$$\underset{\sim}{B}^T = [0.0013 \quad 0.0539]$$

$$\underset{\sim}{H} = \underset{\sim}{0}$$

$$\underset{\sim}{Q} = \begin{bmatrix} 0.25 & 0.0 \\ 0.0 & 0.05 \end{bmatrix}$$

$$R = 0.05$$

Order of the system, N = 2

Number of inputs, M = 1

For this problem NS is chosen to be 200 and the computer control and data cards are

```
// (standard OS JOB card)
//‸EXEC‸LINCON
//LINK.SYSIN‸DD‸*
‸‸INCLUDE SYSLIB(STREG)
/*
//GO.SYSIN‸DD‸*
```

```
STREG  TEST    0201
0.9974    0.0539
-.1078    1.1591
0.0013    0.0539
0.0       0.0
0.0       0.0
0.25      0.0
0.0       0.05
0.05
200
/*
```

The results presented in fig. 3-47 show that $\underset{\sim}{F}(NS = K)$ approaches a constant matrix[5] $\underset{\sim}{F}$ as $K \rightarrow 200$.

10. Multiple-Input Multiple-Output Control System Decoupling (MIMO)

This subprogram is used to determine a feedback con- control law

$$\underset{\sim}{u}(t) = \underset{\sim}{G}\,\underset{\sim}{r}(t) + \underset{\sim}{F}\,\underset{\sim}{x}(t)$$

[5] If a system is completely controllable and time invariant, $\underset{\sim}{H} = \underset{\sim}{0}$, and $\underset{\sim}{R}$ and $\underset{\sim}{Q}$ are constant matrices then [12]

$$\underset{\sim}{F}(NS - K) \rightarrow \underset{\sim}{F} \text{ (a constant matrix) as } NS \rightarrow \infty$$

```
DISCRETE LINEAR STATE REGULATOR PROGRAM
PROBLEM IDENTIFICATION = STREG TEST
**********************************************

 THE A MATRIX

9.57400000D-01        5.39000000D-02
-1.07800000D-01       1.15910000D 00

 THE B MATRIX

1.30000000D-03        5.39000000D-02

 THE H MATRIX

0.0                   0.0
0.0                   0.0

 THE Q MATRIX

2.50000000D-01        0.0
0.0                   5.00000000D-02

 THE R MATRIX

5.00000000D-02
**********************************************

 K = 199
GAINS
0.0                   0.0

 K = 198
GAINS
-6.70725733D-04       -6.26433187D-02

 K = 197
GAINS
-6.93244869D-03       -1.47394414D-01

 K = 196
GAINS
-1.66484130D-02       -2.61281131D-01

 K = 195
GAINS
-2.63789511D-02       -4.12206705D-01

 K = 194
GAINS
-3.12184177D-02       -6.08210446D-01

 K = 193
GAINS
-2.49835494D-02       -8.56139598D-01

 K = 192
GAINS
-2.38525194D-04       -1.15972835D 00

 K = 191
GAINS
4.56129417D-02        -1.51739895D 00

 K = 190
GAINS
1.29745646D-01        -1.92052614D 00

 K = 189
GAINS
2.41754156D-01        -2.35316543D 00

 K = 188
GAINS
3.82699159D-01        -2.79358045D 00

 K = 187
GAINS
5.45256077D-01        -3.22004230D 00

 K = 186
GAINS
7.19140314D-01        -3.61130956D 00

 K = 185
GAINS
8.93241929D-01        -3.95399295D 00

 K = 184
GAINS
1.05762030D 00        -4.24157675D 00

 K =  14
GAINS
-5.52229654D-01       -5.96901509D 00

 K =  13
GAINS
-5.52229654D-01       -5.96901509D 00

 K =  12
GAINS
-5.52229654D-01       -5.96901509D 00

 K =  11
GAINS
-5.52229654D-01       -5.96901509D 00

 K =  10
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   9
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   8
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   7
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   6
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   5
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   4
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   3
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   2
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   1
GAINS
-5.52229654D-01       -5.96901509D 00

 K =   0
GAINS
-5.52229654D-01       -5.96901509D 00
**********************************************

AND THE P(200) MATRIX IS

     1.65226263D 01        1.01738418D 00
     1.01738418D 00        6.50995648D 00
```

Figure 3-47 Discrete Linear State Regulator Test

for an Nth order system

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{B}\,\underset{\sim}{u}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

such that the control system is decoupled, i.e., the i^{th} input $r_i(t)$ affects only the i^{th} output $y_i(t)$. Notice that the subprogram applies only if the number of inputs is exactly the same as the number of outputs. The computer calculates both the feedback gain matrix $\underset{\sim}{F}$ and the command input gain matrix $\underset{\sim}{G}$. The user only has to feed in the coupled system matrices $\underset{\sim}{A}$, $\underset{\sim}{B}$ and $\underset{\sim}{C}$ and specify, arbitrarily, the desired closed-loop poles of each i^{th} transfer function $Y_i(s)/R_i(s)$.

The theory regarding the algorithm used for decoupling is not presented here. For this, the reader is referred to [1]. Sufficient information is included, however, to illustrate the concepts and to permit easy use of the subprogram.

a. Input

The problem identification, the system order (N less than or equal to ten) and the number of inputs and outputs (M less than or equal to ten) are given on the first card according to the format shown on the input format table for MIMO. Then the $\underset{\sim}{A}$ matrix (N×N) is entered, followed by the $\underset{\sim}{B}^T$ matrix (M×N) and the $\underset{\sim}{C}$ matrix (M×N), one row at a time. Note that $\underset{\sim}{B}$ is transposed and the number of inputs must equal the number of outputs.

Next the option card is punched. If the option is blank, the phase variable form of the decoupled system is obtained and the subprogram returns to begin another problem. If options P or F are selected, the control law $\underset{\sim}{u}(t)$ necessary to achieve a decoupled system with closed-loop poles at locations specified by the user is determined. If option = F, the next cards give the desired poles of $Y_1(s)/R_1(s)$.

According to the convention established before, if option F is selected the real part of a root is entered as being positive if it lies in the left-half plane, negative if in the right-half plane and only the positive imaginary part of a complex pair is given (see p.). If option P is selected, the coefficients of the characteristic polynomial of $Y_1(s)/R_1(s)$ are entered in ascending order, the coefficient of the highest order term always being unity.

The subprogram then returns to read option P or F and the second decoupled subsystem desired closed-loop poles, and so on for the M subsystems.

The design of a decoupled system requires two separate runs of the subprogram. First the user must determine if it is possible to decouple the system and, if so, obtain the order of each decoupled subsystem $Y_i(s)/R_i(s)$. This is done by leaving the option card blank. The order of the denominator polynomial becomes the order of each decoupled subsystem which determines the number of poles or the degree of the characteristic polynomial to be selected for closed-loop calculations.

Options P or F are selected for the second run and the subprogram computes the control law $\underset{\sim}{u}(t)$ which decouples the system and places the poles at the selected locations. The following input format table summarizes the pertinent information.

Entry	Input Description	Format	Columns Used
1	Problem identification, order of the system ($N \leq 10$), number of inputs and number of outputs ($M \geq 10$)	5A4, 2I2	1-10, 21-22, 23-24
2	$\underset{\sim}{A}$ matrix (N×N) (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
3	$\underset{\sim}{B}^T$ matrix (M×N) (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.
4	$\underset{\sim}{C}$ matrix (M×N) (one row per card for $N \leq 8$; one row per two cards for $N > 8$)	8F10.3	1-10, 11-20, 21-30, etc.

Entry	Input Description		Format	Columns Used
5 option	Option	blank = analysis only P = closed-loop polynomial input F = closed-loop poles input	A1	1
6 (iff option =P)	Polynomial coefficients in ascending power of s (see complete description p. 31)		8F10.3	1-10, 11-20, 21-30, etc.
7 (iff option =F)	Roots of characteristic polynomial (one root per card) (see complete description p. 32)		8F10.3	1-10, 11-20, 21-30, etc.

The above information should become clear from the example presented in part c.

b. Output

The problem identification, $\underset{\sim}{A}$, $\underset{\sim}{B}^T$ and $\underset{\sim}{C}$ are listed for reference. Then the decoupled phase variable representation of each subsystem is printed. The denominator polynomial in ascending powers of s is given first, followed by the numerator polynomial both in unfactored and factored form. It should be noted that the subprogram outputs the cancelled zeros of $Y_i(s)/R_i(s)$ as well.

If closed-loop calculations have been requested, by letting option equal P or F, each subsystem closed-loop polynomial is printed again both in unfactored and factored

form. Finally the feedback gain matrix $\underset{\sim}{F}$ and the control gain matrix $\underset{\sim}{G}$ are presented.

In terms of the original system, the resulting closed-loop decoupled system is

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A}\,\underset{\sim}{x}(t) + \underset{\sim}{B}\,\underset{\sim}{u}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

$$\underset{\sim}{u}(t) = \underset{\sim}{F}\,\underset{\sim}{x}(t) + \underset{\sim}{G}\,\underset{\sim}{r}(t)$$

or, in a form suitable for graphical time response simulation,

$$\dot{\underset{\sim}{x}}(t) = (\underset{\sim}{A} + \underset{\sim}{B}\underset{\sim}{F})\,\underset{\sim}{x}(t) + \underset{\sim}{B}\underset{\sim}{G}\underset{\sim}{r}(t)$$

$$\underset{\sim}{y}(t) = \underset{\sim}{C}\,\underset{\sim}{x}(t)$$

It must be pointed out that not every system can be decoupled. If it cannot, the subprogram is interrupted and the message "BSTAR IS SINGULAR - THIS SYSTEM CANNOT BE DECOUPLED" is printed. It is also possible that a subsystem may be uncontrollable. This is indicated as such on the output listing.

c. Example

A two-input two-output system [13] is to be decoupled both during transient-period and steady-state.

The first subsystem must approach a second order response to a step input with a natural frequency of 10, a damping factor of 0.4 and no steady-state error. The second subsystem must also approach a second order response to step input but with a natural frequency of 4, a damping factor of 0.6 and no steady-state error.

The original system is shown both in block diagram and signal flow graph form in Figures 3-48 and 3-49.

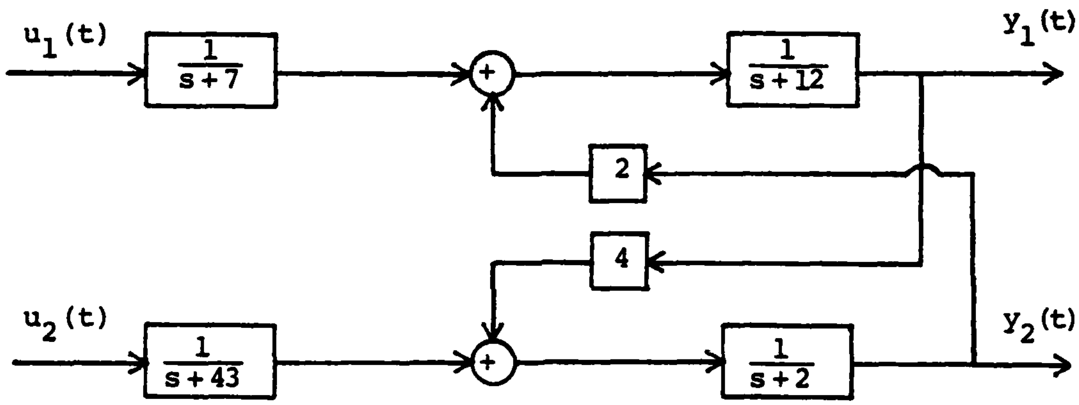


Fig 3-48 Multiple-Input Multiple-Output Control System (Block Diagram)

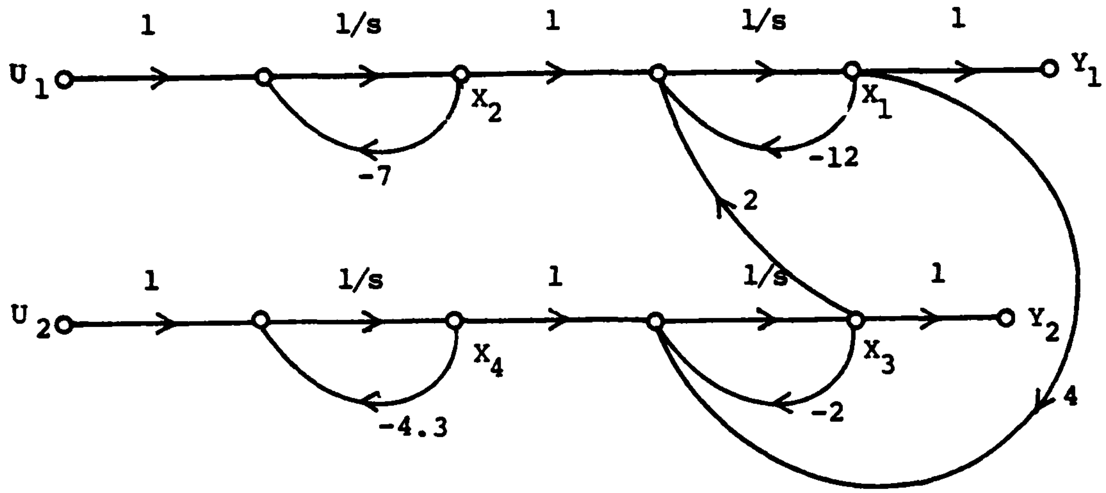


Fig 3-49 Multiple-Input Multiple-Output Control System (Signal Flow Graph)

The state variable and output equations can be directly written as

$$\dot{x}_1 = -12x_1 + x_2 + 2x_3$$

$$\dot{x}_2 = -7x_2 + u_1$$

$$\dot{x}_3 = 4x_1 - 2x_3 + x_4$$

$$\dot{x}_4 = -4.3x_4 + u_2$$

$$y_1 = x_1$$

$$y_2 = x_3$$

From these, the matrices $\underset{\sim}{A}$, $\underset{\sim}{B}^T$ and $\underset{\sim}{C}$ are seen to be:

$$\underset{\sim}{A} = \begin{bmatrix} -12 & 1 & 2 & 0 \\ 0 & -7 & 0 & 0 \\ 4 & 0 & -2 & 1 \\ 0 & 0 & 0 & -4.3 \end{bmatrix}$$

$$\underset{\sim}{B}^T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\underset{\sim}{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

A first run of the subprogram can be made to verify if it is possible to decouple the system and find the order of each subsystem. The computer cards are

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(MIMO)
/*
//GO.SYSIN^DD^*
MIMO TEST ONE    040202
-12.      1.0       2.0       0.0
0.0       -7.0      0.0       0.0
4.0       0.0       -2.0      1.0
```

0.0	0.0	0.0	-4.3
0.0	1.0	0.0	0.0
0.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0

(blank card)

/*

The result shown in Fig. 3-50 reveals that both subsystems are second order. The closed-loop pole locations can then easily be selected for each subsystem. For the first one, a second order response is desired such that $\omega_n = 10$, $\zeta = 0.4$. Thus,

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 8s + 100$$

For the second subsystem, the desired response requires that $\omega_n = 4$ and $\zeta = 0.6$, so

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 4.8s + 16$$

$$= (s + 2.4 + j3.2)(s + 2.4 - j3.2)$$

The computer deck is then modified as follows:

```
// (standard OS JOB card)
//^EXEC^LINCON
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(MIMO)
/*
```

```
     MULTI-INPUT, MULTI-OUTPUT PROGRAM
     PROBLEM IDENTIFICATION -      MIMO TEST ONE
*********************************************
     THE A MATRIX
  -1.200000E 01    1.000000E 00    2.000000E 00    0.0
   0.0            -7.000000E 00    0.0             0.0
   4.000000E 00    0.0            -2.000000E 00    1.000000E 00
   0.0             0.0             0.0            -4.300000E 00
     THE B MATRIX
   0.0             1.000000E 00    0.0             0.0
   0.0             0.0             0.0             1.000000E 00
     THE C MATRIX
   1.000000E 00    0.0             0.0             0.0
   0.0             0.0             1.000000E 00    0.0
*********************************************
     DECOUPLED PHASE VARIABLE REPRESENTATION
**** SUBSYSTEM  1
     DENOMINATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   1.90735E-06    -3.81470E-06    1.00000E 00
     NUMERATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   1.00000E 00
**** SUBSYSTEM  2
     DENOMINATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   0.0            0.0             1.00000E 00
     NUMERATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   1.00000E 00
```

Figure 3-50 Computer Output for MIMO Test One

```
//GO.SYSIN DD *
MIMO TEST TWO      040202
-12.0       1.0       2.0       0.0
0.0         -7.0      0.0       0.0
4.0         0.0       -2.0      1.0
0.0         0.0       0.0       -4.3
0.0         1.0       0.0       0.0
0.0         0.0       0.0       1.0
1.0         0.0       0.0       0.0
0.0         0.0       1.0       0.0
P
100.        8.0       1.0
F
2.4         3.2
/*
```

From the results given in Fig. 3-51, the decoupled compensated system can be written in terms of the original system.

$$\dot{\underset{\sim}{x}}(t) = \underset{\sim}{A} + \underset{\sim\sim}{BF}\ \underset{\sim}{x}(t) + \underset{\sim\sim}{BG}\ \underset{\sim}{r}(t)$$

$$= \begin{bmatrix} -12 & 1 & 2 & 0 \\ -156 & 4 & 12 & -2 \\ 4 & 0 & -2 & 1 \\ 36.8 & -4 & -18.4 & -2.8 \end{bmatrix} \underset{\sim}{x}(t) + \begin{bmatrix} 0 & 0 \\ 100 & 0 \\ 0 & 0 \\ 0 & 16 \end{bmatrix} r(t)$$

```
     MULTI-INPUT, MULTI-OUTPUT PROGRAM
     PROBLEM IDENTIFICATION -      MIMO TEST TWO
*********************************************
     THE A MATRIX
  -1.20000E 01     1.00000E 00     2.00000E 00     0.0
   0.0            -7.00000E 00     0.0             0.0
   4.00000E 00     0.0            -2.00000E 00     1.00000E 00
   0.0             0.0             0.0            -4.30000E 00
     THE B MATRIX
   0.0             1.00000E 00     0.0             0.0
   0.0             0.0             0.0             1.00000E 00
     THE C MATRIX
   1.00000E 00     0.0             0.0             0.0
   0.0             0.0             1.00000E 00     0.0
*********************************************
     DECOUPLED PHASE VARIABLE REPRESENTATION
**** SUBSYSTEM  1
     DENOMINATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   1.90735E-06   -3.81470E-06    1.00000E 00
     NUMERATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   1.00000E 00
**** SUBSYSTEM  2
     DENOMINATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   0.0            0.0            1.00000E 00
     NUMERATOR POLYNOMIAL - IN ASCENDING POWERS OF S
   1.00000E 00
*********************************************
     CLOSED-LOOP CALCULATIONS
*********************************************
**** SUBSYSTEM  1
     CLOSED-LOOP POLYNOMIAL - IN ASCENDING POWERS OF S
   1.00000E 02    8.00000E 00    1.00000E 00
     CLOSED-LOOP POLES             REAL PART          IMAGINARY PART
                                -4.0000000E 00      -9.1651516E 00
                                -4.0000000E 00       9.1651516E 00
**** SUBSYSTEM  2
     CLOSED-LOOP POLYNOMIAL - IN ASCENDING POWERS OF S
   1.60000E 01    4.80000E 00    1.00000E 00
     CLOSED-LOOP POLES             REAL PART          IMAGINARY PART
                                -2.3999996E 00      -3.2000008E 00
                                -2.3999996E 00       3.2000008E 00
*********************************************
     ORIGINAL STATE VARIABLES
     FEEDBACK GAIN MATRIX
  -1.56000E 02     1.10000E 01     1.20000E 01    -2.00000E 00
   3.68000E 01    -4.00000E 00    -1.84000E 01     1.50000E 00
     CONTROL GAIN MATRIX
   1.00000E 02     0.0
   0.0             1.60000E 01
*********************************************
```

Figure 3-51 Computer Output for MIMO Test Two

$$y_1(t) = x_1(t)$$

$$y_2(t) = x_3(t)$$

For comparison with the actual results given in [13], the decoupled compensated system was simulated using the subprogram GTRESP, for a unit step input. Note that since GTRESP only allows for single-input single-output simulation, two runs must be made. The data for the subprogram is

$$r(t) = 1.0$$

$$\underset{\sim}{A} = \begin{bmatrix} -12 & 1 & 2 & 0 \\ -156 & 4 & 12 & -2 \\ 4 & 0 & -2 & 1 \\ 36.8 & -4 & -18.4 & -2.8 \end{bmatrix}$$

$$\underset{\sim}{b}^T = [0 \quad 100 \quad 0 \quad 0] \text{, for the first channel}$$

$$\underset{\sim}{b}^T = [0 \quad 0 \quad 0 \quad 16] \text{, for the second channel}$$

$$\underset{\sim}{c} = [1 \quad 0 \quad 0 \quad 0] \text{, for the first channel}$$

$$\underset{\sim}{c} = [0 \quad 0 \quad 1 \quad 0] \text{, for the second channel}$$

$$\underset{\sim}{k}^T = \underset{\sim}{0}$$

$K = 1$

$\underset{\sim}{x}(t_0) = \underset{\sim}{0}$

TZERO = 0.   TF = 10.

DT = 0.01   FREQ = 20

The output $y_1(t)$ and $y_2(t)$ were to be plotted for the first and second run, respectively. The complete computer deck for GTRESP follows.

```
// (standard OS JOB card),TIME=2
//^EXEC^LINCONF
//FORT.SYSIN^DD^*
        SUBROUTINE RFIND(T,R)
        R=1.0
        RETURN
        END
/*
//LINK.SYSIN^DD^*
^^INCLUDE^SYSLIB(GTRESP)
^^ENTRY^GTRESP
/*
//GO.SYSIN^DD^*
GTRESP MIMO      04
-12.0     1.0     2.0     0.0
-156.0    4.0     12.0    -2.0
```

```
4.0      0.0      -2.0     1.0
36.8     -4.0     -18.4    -2.8
0.0      100.0    0.0      0.0      Note:  these b^T and c
                                           matrices are for
1.0      0.0      0.0      0.0             the first channel
                                           simulation
0.0
1.0
0.0
0.0      10.0     0.01     20.
Y
/*
```

For the second channel, the subprogram is run a second time changing the $\underset{\sim}{b}^T$ and $\underset{\sim}{c}$ matrices appropriately. Figures 3-52A and 3-52B show that the response effectively meets the specifications given initially.

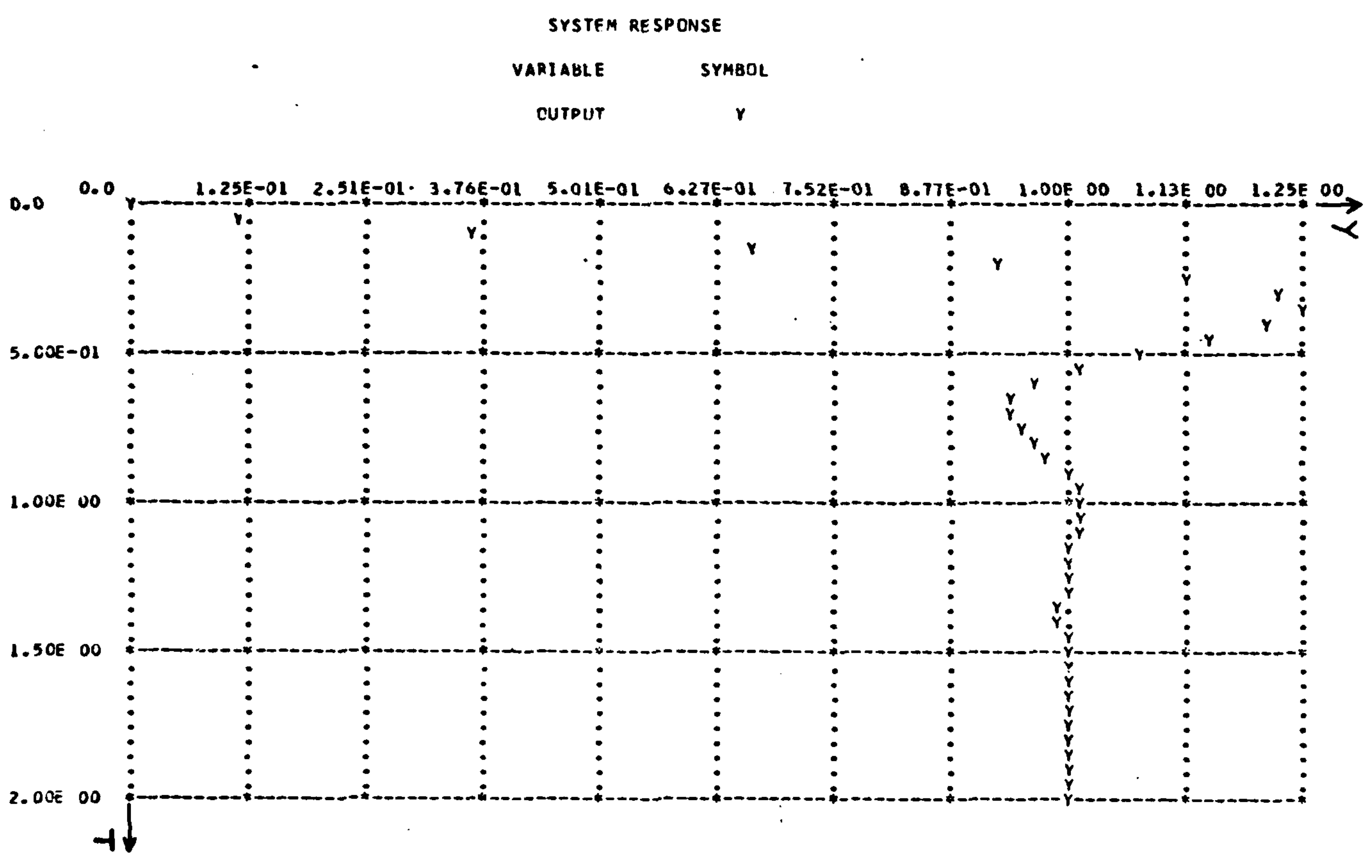


Figure 3-52A GTRESP for Channel One

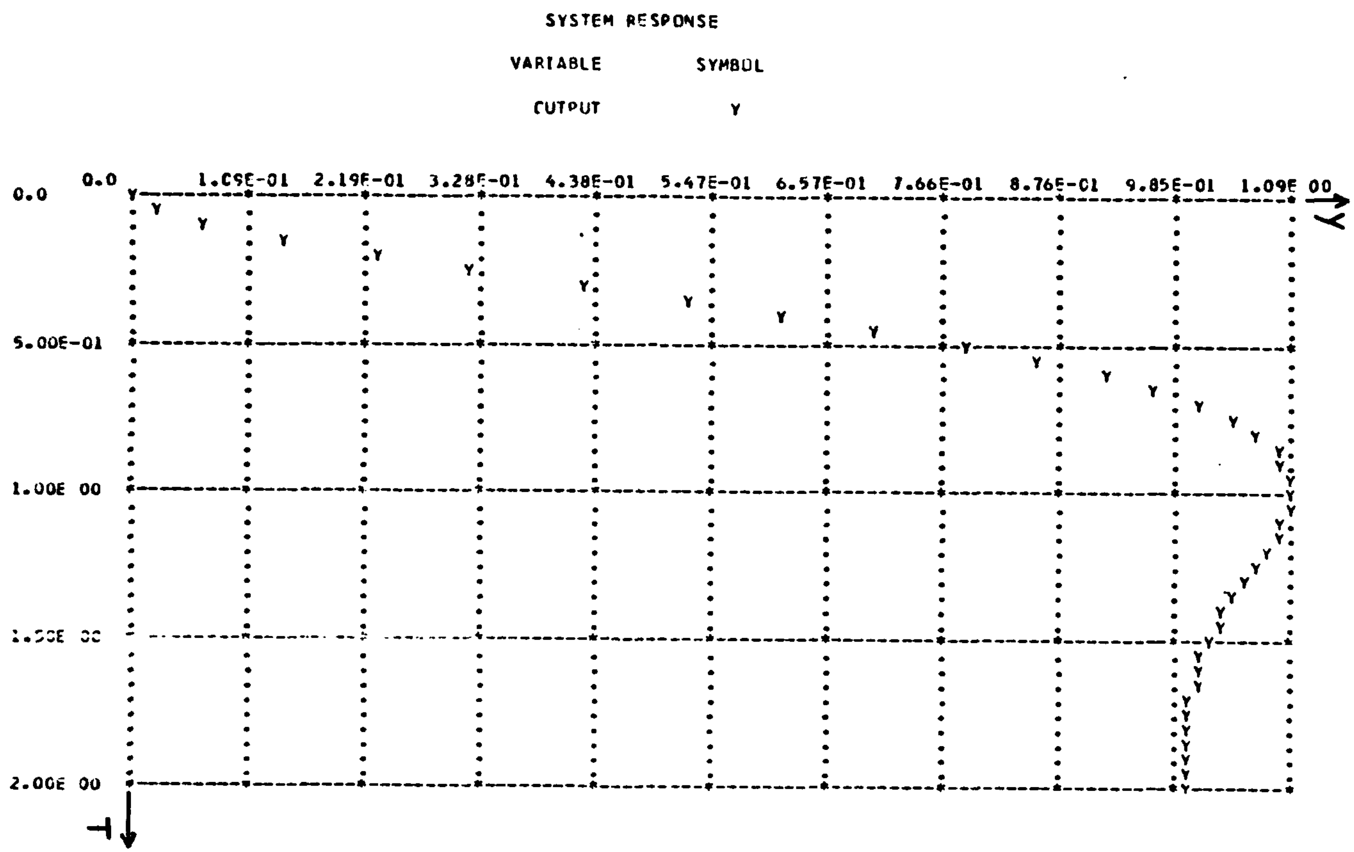


Figure 3-52B GTRESP for Channel Two

IV. CONCLUSIONS AND RECOMMENDATIONS

The eighteen subprograms presented in the third chapter constitute the actual linear control subroutine library and this thesis, the user's manual that goes with it. The subprograms are easy to access and have proven to work well. (They were all tested by solving several textbook problems and, hopefully, all the "bugs" have been eliminated.) Very little programming is necessary so the user can concentrate on control problems rather than worry about computational details. The LINCON library is indeed a nice tool for analysis and design of linear control systems.

Furthermore, the library can still be easily improved and expanded. Any FORTRAN subroutines can be modified and replaced or new subroutines added by following the simple instructions given in Appendix A. Note that, as was done in reference [1], the subprograms were written to handle systems of order less than or equal to ten. This should take care of most of the problems encountered. If, however, it becomes necessary to solve higher order systems using these subprograms, remember that it can easily be accomplished by re-dimensioning the arrays of the appropriate subroutines and replacing them in the load module (again following the procedure given in Appendix A).

Finally, the following recommendations should be taken into consideration

(1) by the users,

- always use the proper job control language cards (i.e., the ones described in Chapters II and III) to access the subprograms.

(2) by the future LINCON library "programmer",

- before making any changes, be certain the control cards are exactly those required for the job. Double checking with a consultant is always a good idea.
- always keep a copy of the subroutines' listings and the card decks. It is not possible to obtain any listings or card decks from a load module.
- every modification should be documented (complete with examples) and the information distributed to the users.
- after changes have been implemented and tested, a back-up copy of the new LINCON data sets should be created to replace the one on magnetic tape (as specified in Appendix A).

APPENDIX A

The LINCON Data Sets

All the linear control subprograms described in Chapter III were placed in a load module (pre-processed by the linkage editor) on Disk 02 of the Naval Postgraduate School OS/MVT IBM/360. Procedures were cataloged so anyone can easily access the subprograms under OS Batch. The job control language cards to be prepared to use the load module linear control library are given in Chapter II.

The following paragraphs now present the actual content of the load module and explain the procedures to

- modify or add members
- change the data set's expiration date
- delete the data sets
- list the member names and check the disk space
- compress the data sets.

Also, since a back-up copy of the data sets was created, the procedure to restore the load module linear control library is given as well.

However, before any attempt is made to "play" with the load module, it is suggested that the programmer familiarize himself with the latest computer procedures and the linear control subroutine library. References 14 and 15 should also be read.

1. Content of The Library.

The linear control subroutines library contains a total of fifty-two subroutines. These are given in Table A-1 indicating what subroutines are used by the subprograms. A detailed description of most of the subroutines is presented in [1]. Note that several minor changes had to be made to the subroutines in order to implement the library. These modifications do not, however, change the purpose or the efficiency of the programs. Anyone interested in the programming aspects of the library must utilize both the subroutine listings and reference [1].

2. Data Sets Utilities

It is probable that the content of the LINCON subroutines library will have to be modified at one time or another. The following paragraphs outline the procedure and give the job control language cards necessary to carry out the changes.

a. Data Set Listing

The following set of control cards is used to list the load module library content and the spaces it occupies:

```
//  (standard OS JOB card)
//^ EXEC^PGM=IEHLIST
//SYSPRINT^ DD ^SYSOUT=A
//DD1^ DD ^UNIT=3330,VOL=SER=DISK02,DISP=SHR
//SYSIN ^DD ^*
^ ^ LISTVTOC ^FORMAT,VOL=3330=DISK02,DSNAME=F0718.LINCON
^ ^ LISTPDS ^ VOL=3330=DISK02,DSNAME=F0718.LINCON
/*
```

	BASMAT	CONOBS	FRESP	GTRESP(*)	KALMAN(*)	LUEN	MAIN (**)	MIMO	OBSERV	PRFEXP	PRTLOC	RICATI(*)	ROOTS	RTLOC	RTRESP	SENSIT	SERCOM	STRGG	STVAR
CALCU				x															
CHREQ	x															x			
CHREQA	x							x							x	x			
DET	x							x							x	x			x
DIVP										x									x
DMULT					x													x	
FORM																x			
GRAPH			x																
HERMIT		x						x	x										
LINEQ						x											x		
MAXI			x																
MPY																x			
MULT		x						x	x										
NORMP										x									
PADD										x									
PEXCG										x									
PFEXP										x									
PHNOM			x																
PMUL										x									
POLRT													x						
PROOT	x		x			x			x	x	x			x	x	x		x	
PVAL			x							x									
RUNGE				x															
SEMBL			x			x		x		x				x	x				
SIMEQ	x							x											x
SIMUL					x							x						x	
SORT										x									
SPLIT			x							x				x		x			
STMST	x														x				
SUBP										x									
TRESP				x															
VECTEQ													x						
YDOT				x															
Y8VSX				x															

Table A-1 Subroutines Cross List

(*) These subprograms were loaded with all their necessary subroutines. Each one of them requires an external subroutine (see Chapter III).

(**) The subprogram MAIN is used to call all the subprograms (operation under Mode Three). It requires 450K core.

All the subprograms and subroutines names are listed in alphabetical order and the space occupied and unoccupied given in terms of number of tracks and number of cylinders.

b. Changing Expiration Date

The expiration date of the subroutine library must be changed approximately every six months. The control cards used to perform the task are:

```
//  (standard OS JOB card)
//^EXEC^PGM=CEXPDATE
//SYSPRINT^DD^SYSOUT=A
//DD1^DD^UNIT=3330,VOL=SER=DISK02,DSIP=OLD,DSN=F0718.LINCON,
//^LABEL=EXPDT=yyddd
/*
```

where yy=year (e.g. 80)

and ddd=day (e.g. 365)

The last expiration date given was 80182, i.e. 01 July 1980. The computer centre normally sends a reminder listing the data sets that are about to expire.

c. Adding New Members or Replacing Existing Ones

The following control cards are required to add a new member or replace an existing one:

```
//  (standard OS JOB card)
//^EXEC^FORTCL,PARM.LINK='NCAL,MAP,LIST'
//FORT,SYSIN^DD^*
          Subroutine to be modified or added
/*
```

```
//LINK.SYSLMOD DD UNIT=3330,VOL=SER=DISK02,DISP=SHR,
// DSN=F0718.LINCON(member)
/*
```

where "member" is the name of the subroutine to be modified or added. Note that the complete set of cards representing the subroutine called "member" must be included. Before placing the subroutine in the load module library, the computer compiles it. If any error is found, the linkage is not executed and the new subroutine is not placed in the load module. The user must carefully check the computer output and make sure the message "member now replaced in data set" is printed. If not, he must correct any error and redo the procedure. Note that a lack of space can also prevent the computer from linking to the load module. If this last situation occurs the user should run the "data set listing" (part a) control cards to see how much space is available. If sufficient space can be allocated, he must run the "compressing data sets" control cards (part e) to release any unused space in the data sets and then execute the addition or replacement.

d. Removing Data Sets

It is sometimes necessary to remove undesired members from the library (to create space or erase useless programs). The following control cards are used to delete one or several members from the subroutine library:

```
//  (standard OS JOB card)
//^ EXEC^ PGM=IEHPROGM
//SYSPRINT^ DD^ SYSOUT=A
//DD1^ DD^ UNIT=3330,VOL=SER-DISK02,DISP=SHR
//SYSIN^ DD^ *
^ ^ SCRATCH^ VOL=3330=DISK02,PURGE,DSNAME=F0718.LINCON,MEMBER=member1
^ ^ SCRATCH^ VOL=3330=DISK02,PURGE,DSNAME=F0718.LINCON,MEMBER=member2
/*
```

where member1 and member2 are the subroutines to be erased from the module. Here the programmer must be extremely careful while using this utility. Mistakes can be very costly (from scratching the wrong subroutine to erasing the whole subroutine library). For instance, using

```
    SCRATCH VOL=3330=DISK02,PURGE,DSNAME=F0718.LINCON
```

would erase the entire LINCON subroutine library. Be careful.

Also note that scratching a member does not make the space it occupied immediately available. The "compressing data sets" utility must be run to release the space (see part e).

e. Compressing Data Sets

The following control cards are used to free unavailable space in the data set:

```
//  (standard OS JOB card)
//^ EXEC^PGM=IEBCOPY,REGION=100K
//SYSPRINT^ DD^ SYSOUT=A
//DD1^ DD ^UNIT=3330,VOL=SER-DISK02,DSN=F0718.LINCON,DISP=OLD
//SYSUT3^ DD ^UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
```

```
//SYSUT4 ^ DD ^UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSIN ^DD ^*
^ ^ COPY ^ OUTDD=001,INDD=DD1
/*
```

Note that the use of this utility is somewhat dangerous since a power failure or a machine check during compression will make the data set unaccessible by any program [13].

3. Back-up Copy

A back-up copy of the partitioned data sets was made by copying them onto the magnetic tape NPS 705, file 01. The control cards that were used to create it are:

```
//  (standard OS JOB card)
//ONE ^EXEC ^ PGM=IEHMOVE,REGION=80K
//SYSPRINT ^ DD ^ SYSOUT=A
//SYSUT1 ^ DD ^UNIT=SYSDA,SPACE=(CYL,(3,1))
//DDX ^DD ^UNIT=3330,VOL=SER=DISK02,DISP=SHR
//TAPE ^DD ^UNIT=3400-3,VOL=SER=NPS705,DISP+(,PASS),DCB=DEN=3
//SYSIN ^ DD ^ *                                          in column 72
                                                                ↓
 COPY ^ PDS+F0718.LINCON,TO=3400-3=(NPS705,1),                  X
               FROM=3330=DISK02,TODD=TAPE
               ↑
/*          in column 16
```

Since it is possible that the data sets may be lost one way or another, it is imperative to have such a back-up copy. To restore the LINCON data sets, the programmer must first re-allocate space by running the following job control cards:

```
//  (standard OS JOB card)
//TWO^ EXEC ^PGM=IEFBR14
//DDL ^ DD ^UNIT=3330,VOL=SER=DISK02,DISP=(NEW,KEEP),
// ^^DSN=F0718.LINCON,LABEL=EXPDT=yyddd,SPACE=(Cyl,(2,1,10))
/*
```

where yy=expiration year

ddd=expiration day

Finally, to restore the data sets one only has to run the program given below.

```
//  (standard OS JOB card)
//THREE ^ EXEC ^ PGM=IEHMOVE,REGION=80K
//SYSPRINT ^ DD ^ SYSOUT=A
//SYSUT1 ^ DD ^ UNIT=SYSDA,SPACE=(CYL,(3,1))
//DDX ^ DD ^ UNIT=3330,VOL=SER=DISK02,DISP=SHR
//TAPE ^ DD ^ UNIT=3400-3,VOL=SER-NPS705,DISP=(OLD,PASS),DCB=DEN=3
//SYSIN ^ DD ^ *                                         column 72
                                                             ↓
^^  COPY^ PDS=F0718.LINCON,TO=3330=DISK02,                   X
               FROM=3400-3=(NPS705,1),FROMDD=TAPE
               ↑
/*          column 16
```

APPENDIX B

List of the Sources for the Examples of Chapter III

The following table lists the references from which the examples worked out in Chapter II originated.

Section	Example	Reference
IIIB	1.c	Eveleigh, V.W., Introduction to Control System, p. 568 (#3), McGraw-Hill, 1972.
	2.c	Shinners, S.M., Modern Control System - Theory and Application, 2nd ed., p. 364 (#7.26), Addison-Wesley, 1978.
	3.c	Brogan, W.L., Modern Control Theory, p. 35 (#2.10), Quantum Publishers, 1974.
	3.d	Ogata, K., Modern Control Engineering, p. 517, Prentice-Hall, 1970
IIIC	2.c	Ogata, K., Modern Control Engineering, p. 275, Prentice-Hall; 1970
	3.c	Kirk, D.E., Optimal Control Theory - An Introduction, pp. 34-42, Prentice-Hall, 1970.
IIID	1.c	Ogata, K., Modern Control Engineering, p. 797, Prentice-Hall, 1970.
	2.c	Kirk, D.E., Optimal Control Theory - An Introduction, p. 28, Prentice-Hall, 1970.
	3.c	Ogata, K., Modern Control Engineering, pp. 728-729, Prentice-Hall, 1970.
	4.c	Eveleigh, V.W., Introduction to Control System Design, pp. 353-356, McGraw-Hill, 1972.

Section	Example	Reference
	5.d	Eveleigh, V.W., Introduction to Control System Design, pp. 357-360, McGraw-Hill, 1972.
	6.d(1)	Chen, C.T., Introduction to Linear System Theory, p. 296, Holt, Rinehart and Winston, 1970.
	6.d(2)	Same as 5.d
	7.c(1)	Kirk, D.E., Optimal Control Theory - An Introduction, p. 41, Prentice-Hall, 1970.
	7.c(2)	Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, pp. 347-351, Wiley-Interscience, 1972.
	8.c	Parker, S.R., Digital Control Systems (Class Notes), 1978.
	9.c	Kirk, D.E., Optimal Control Theory - An Introduction, Prentice-Hall, 1979.
	10.c	Mowrey, J.T., Compensator Optimization in Multiple Input Multiple Output Control Systems, pp. 26-27, Master's Thesis, Naval Postgraduate School, Monterey, 1979.

Table B-1 List of References for the Examples Worked in Chapter III

REFERENCES

1. Melsa, J.L. and Jones, S.K., Computer Programs for Computational Assistance in the Study of Linear Control Theory, 2nd ed., McGraw-Hill, 1973.

2. IBM System/360 Operating System: Messages and Codes, GC28-6631.

3. Shinners, S.M., Modern Control System - Theory and Application, 2nd ed., Addison-Wesley, 1978.

4. Luenberger, D.G., "Observing the State of a Linear System," IEEE Transactions on Military Electronics, Vol. Mil-8, pp. 74-80, April 1964.

5. Schultz, D.G. and Melsa, J.L., State Functions and Linear Control Systems, McGraw-Hill, 1967.

6. Eveleigh, V.W., Introduction to Control System Design, McGraw-Hill, 1972.

7. Luenberger, D.G., "OBservers for Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-11, pp. 190-197, April 1966.

8. Ferguson, J.D. and Rekasius, Z.V., "Optimal Linear Control Systems with Incomplete State Measurements," IEEE Transactions on Automatic Control, Vol. AC-14, No. 2, April 1969.

9. Athans, M. and Falb, P.F., Optimal Control, McGraw-Hill, 1966.

10. Kirk, D.E., Optimal Estimation: An Introduction to the Theory and Applications, 1975.

11. Parker, S.R., Digital Control System Course (Class Notes), 1978

12. Kirk, D.E., Optimal Control Theory - An Introduction, Prentice-Hall, 1970.

13. Mowrey, J.T., Compensator Optimization in Multiple Input Multiple Output Control Systems, Master's Thesis, Naval Postgraduate School, Monterey, 1979.

14. Naval Postgraduate School Technical Note No. 0141-05, User Libraries and Source Code Editing Under OS, by S.D. Raney, Revised July 1977.

15. Brown, G.D., System/370 Job Control Language, Wiley, 1977.

16. Brogan, W.L., Modern Control Theory, Quantum Publishers, 1970.

17. Ogata, K., Modern Control Engineering, Prentice-Hall, 1970.

18. Chen, C.T., Introduction to Linear System Theory, Holt, Rinehart and Winston, 1970.

19. Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, Wiley-Interscience, 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	15
4.	Professor Alex Gerba, Jr., Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Professor Robert D. Strum, Code 62St Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Cdr. Robert C. Spencer, Code 62Sp Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	Professor George J. Thaler, Code 62Tn Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Professor Harold A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
9.	Professor D.J. Collins, Code 67Co Naval Postgraduate School Monterey, California 93940	1
10.	Professor Robert H. Nunn, Code 66Nn Naval Postgraduate School Monterey, California 93940	1

11. Capt. Berthier A. Desjardins 1
Riviere - 3 - Pistoles
Prov. Quebec, Canada
GOL2EO

ILMED

8